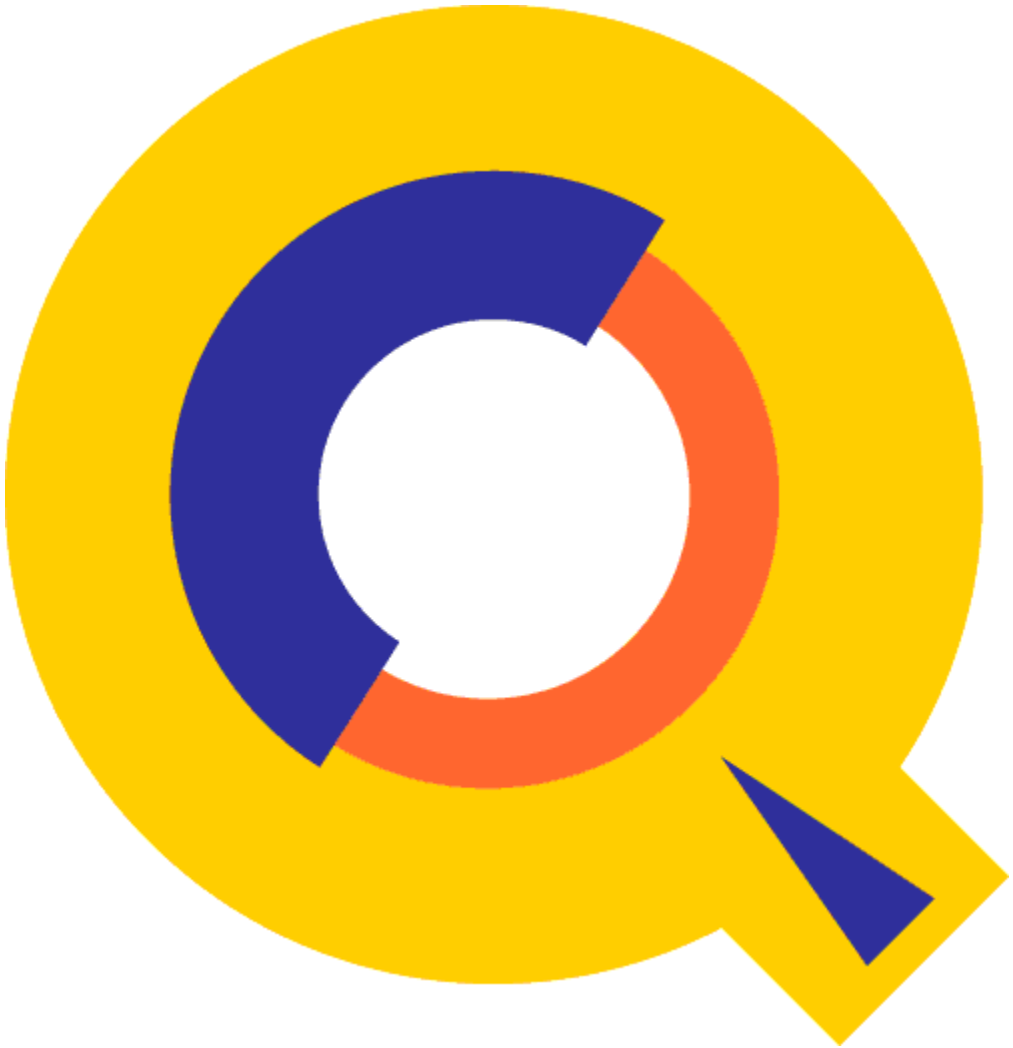
Qedit 5.5 for HP-UX

Change Notice

by Robelle Solutions Technology Inc.



Program and manual copyright © 1977-2004 Robelle Solutions Technology Inc.

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

Updated Tuesday, April 27, 2004

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.

7360 – 137 Street, Suite 372

Surrey, B.C. Canada V3W 1A3

Phone: 604.501.2001

Fax: 604.501.2003

E-mail: sales@robelle.com

E-mail: support@robelle.com

Web: www.robelle.com

Contents

Introducing Qedit/UX 5.5 1

Introduction 1
Qedit for Windows 1
Highlights In Version 5.5 1
Highlights In Version 5.4.12 2
Highlights In Version 5.4.11 2
Highlights In Version 5.4.10 3
Compatibility 3
CPU Serial Number (uname) 4
Known Problems 4
Documentation 4
    User Manual 4
    Change Notice 4
    Printed Documentation 4
    Online Documentation in HTML Help Format 4
    CD or Web Download 5

Installation 7

Introduction 7
Installation Assistance 7
Step 1: Log On as Root 7
Step 2: Create Robelle Directory 8
Step 3: Restore Files 8
Step 4: Setting Up Variables 8
Step 5: Setting Up Links 9
Step 6: Start the Qedit for Windows Server (optional) 9

Enhancements Since Version 5.4 11

Introduction 11
Continuous Printing and Page Ejects 12
Scrollup Character and Function 13
Merge Lines Horizontally 13
Preserving Trailing Spaces 14
Variable Substitution 15
Temporary Files 15
Invalid Line Numbers 16
Sorting Lines 16
Cobol File Extension 17
Form Command 17
Colcopy Command [COL] 18
Colmove Command [COLM] 20
Excluding Lines From Visual Mode Display 22
    Set Visual XX 23

Qedit for Windows 25

Introduction....25
Server Process....25
HP-UX 10.20/11.0....25
Port Number....26

Bugs Fixed 29

Bugs Fixed Since Version 5.4....29

Introducing Qedit/UX 5.5

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 5.5

- Setting a permanent redo stack (**Set Redo**) is ignored when Qedit is running in server mode.
- **Set List Lines 999** skips to a new page only on $page directives.
- **Colcopy** and **Colmove** commands inserted invalid characters when the source columns exceeded the current line length.
- An **Undo** command sometimes reported an error when the **TrailingSpaces** option was enabled.

Highlights In Version 5.4.12

This is an overview of all the changes implemented in this version.

- The Merge-justified (**MergeJ**) command merges the content of a file with the current workfile based on the line numbers by appending text of the external file to the corresponding line in the workfile.
- You can define a scrollup character with **Set Visual Scrollup** to be used in the copy/paste columns of full-screen mode. The default character is a minus sign "-". When used, Qedit scrolls up (backward towards the start of the file) by the number of lines configured in the **Set Visual Roll** item. You can enter as many scrollup characters as needed to go back that many **Roll** values.
- Certain monitors do not support halfbright display enhancements very well. Some messages and prompts are hardly visible. To prevent Qedit from using halfbright, enter **Set Halfbright Off**.
- Trying to copy/move a block of lines where the start or end of the block is on a newly inserted line (relative number not assigned yet) in full-screen mode returned an error message: Copy/paste functions not allowed on Excluded Area line.
- Inserting new lines in full-screen mode which caused a background renumbering operation caused corruption (lines inserted in the wrong place, duplicate line numbers).
- Interrupting a listing with Control-C (interrupt signal equivalent to Control-Y on MPE) after exiting from Screen mode does not cause an abort anymore.

Highlights In Version 5.4.11

This is an overview of all the changes implemented in this version.

- The Lsort command can be abbreviated to "lso", "lsq" (quiet), "lst" (template) and "lsj" (justify). "ls" followed by a space executes the HP-UX shell command. "ls" followed by any other character is executed as a possible shell command.
- Qedit can substitute environment variable names such as `$HOME`, `$PWD` used in commands. This feature needs to be enabled using **Set Varsub On**.
- By default, Qedit now preserves trailing spaces from all lines in a variable-length file. Qedit was unable to write odd-length records. Odd-length lines always ended up with an extra space making the line even-length. **Set Work Trailingspaces** now allows control over trailing spaces and fixes the problem with odd-length lines. When the option is enabled (default), trailing spaces become significant and are preserved.
- An invalid line number is now displayed on the appropriate error message. This should help users identify which line is causing the error.
- Qedit displays an error message if you try to use it as a Qedit for Windows server (-d argument) but the server module has not been enabled. In this case, the Qedit for Windows client is notified and displays an appropriate error message.

- There were a number of problems with random names generated for scratchfiles and temporary Hold files. These names can now have up to 240 characters and are referred to by their absolute pathnames.
- Running Qedit with Blockemulation mode enabled does not work with QCTerm 1.02 or older. It works on version 2.0 and higher. QCTerm is a free terminal emulator from AICS Research.
- Full-screen did not always change the display width automatically in Qcterm.
- The Qedit for Windows client sometimes aborted with a linecount discrepancy error after a paste operation. This was caused by a problem in the server.

Highlights In Version 5.4.10

These are the changes made to Qedit since the last major release (version 5.4).

- The **Lsort** command is now available in Qedit/UX.
- The `.pco` extension is recognized as a valid Cobol source. This extension is typically used to identify Cobol source files that needs to be processed by the Oracle pre-compiler.
- The server and client views of a file were sometimes different after pasting some lines in a Cobol source file. The problem could lead to file corruption, strange behavior and even client aborts.
- The server notifies the Qedit for Windows client whenever it runs out of line numbers (Error: Too High). Qedit for Windows displays an appropriate message and the file is still opened but it may be missing some lines.
- The new **Form** command displays information about self-describing files created with programs such as Suprtool.
- It's now possible to copy or move columns to a different location on a line. The new commands are respectively called **Colcopy** and **Colmove.**
- Full-screen mode now offers an eXclude feature. You can mark a line range for exclusion so it's not displayed in full-screen. All the marked lines are replaced with a single marker. Lines can be excluded using the **Set Visual XX** command or by using **XX** indicators in the cut-and-paste area in full-screen mode.

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 available in Screen mode or by enabling the **Blockemulation** option on HP-UX 11.0 and later.

Qedit commands are added to the Redo stack before variable substitution occurs (**Set Varsub On**) i.e. with the variable name. So, if the variable value changes between the time the command is entered and the time it is retrieved from the stack, the results may be different. It's also important to note that commands related to

Redo stack operations such as **Listredo**, **Do**, **Before** can not have trailing comments enclosed in curly braces anymore. The comments are not removed and likely cause a syntax error.

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 no known problems at the moment.

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 Since 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.

- Set List Lines 999 skips to a new page only on $page directives. [5.4.13]
- Setting a permanent redo stack (**Set Redo**) is ignored when Qedit is running in server mode. [5.4.13]
- The Merge-justified (**MergeJ**) command merges the content of a file with the current workfile based on the line numbers by appending text of the external file to the corresponding line in the workfile. [5.4.12]
- You can define a scrollup character with **Set Visual Scrollup** to be used in the copy/paste columns of full-screen mode. The default character is a minus sign "-". When used, Qedit scrolls up (backward towards the start of the file) by the number of lines configured in the **Set Visual Roll** item. You can enter as many scrollup characters as needed to go back that many **Roll** values. [5.4.12]
- Certain monitors do not support halfbright display enhancements very well. Some messages and prompts are hardly visible. To prevent Qedit from using halfbright, enter **Set Halfbright Off**. [5.4.12]
- The Lsort command can be abbreviated to "lso", "lsq" (quiet), "lst" (template) and "lsj" (justify). "ls" followed by a space executes the HP-UX shell command. "ls" followed by any other character is executed as a possible shell command. [5.4.11]
- Qedit can substitute environment variable names such as `$HOME`, `$PWD` used in commands. This feature needs to be enabled using **Set Varsub On**. [5.4.11]
- By default, Qedit now preserves trailing spaces from all lines in a variable-length file. Qedit was unable to write odd-length records. Odd-length lines always ended up with an extra space making the line even-length. **Set Work Trailingspaces** now allows control over trailing spaces and fixes the problem with odd-length lines. When the option is

enabled (default), trailing spaces become significant and are preserved. [5.4.11]

- An invalid line number is now displayed on the appropriate error message. This should help users identify which line is causing the error. [5.4.11]
- Qedit displays an error message if you try to use it as a Qedit for Windows server (-d argument) but the server module has not been enabled. In this case, the Qedit for Windows client is notified and displays an appropriate error message. [5.4.11]
- There were a number of problems with random names generated for scratchfiles and temporary Hold files. These names can now have up to 240 characters and are referred to by their absolute pathnames. [5.4.11]
- There were a number of problems with random names generated for scratchfiles and temporary Hold files. These names can now have up to 240 characters and are referred to by their absolute pathnames. [5.4.11]
- The **Lsort** command is now available in Qedit/UX. [5.4.10]
- The `.pco` extension is recognized as a valid Cobol source. This extension is typically used to identify Cobol source files that needs to be processed by the Oracle pre-compiler. [5.4.10]
- The new **Form** command displays information about self-describing files created with programs such as Suprtool. [5.4.10]
- It's now possible to copy or move columns to a different location on a line. The new commands are respectively called **Colcopy** and **Colmove.** [5.4.10]
- Full-screen mode now offers an eXclude feature. You can mark a line range for exclusion so it's not displayed in full-screen. All the marked lines are replaced with a single marker. Lines can be excluded using the **Set Visual XX** command or by using **XX** indicators in the cut-and-paste area in full-screen mode. [5.4.10]

Continuous Printing and Page Ejects

You can control the output of a **List** command using the **Lines** option among others. This option can be set with **Set List Lines** (global default) or using the **$Lines** option on the List command itself.

A number between 1 and 256 specifies the maiximum number of lines on each page. Qedit skips to a new page every time the specified number of lines is reached or when a $page directive is encountered in the text. You can prevent page ejects completely by entering **Lines 0**.

To perform continuous printing with no automatic page ejects but skip to a new page on $Page directives, use **$lines 999**.

```
/list $lp $lines 999 all     {skips to a new page on $page only}
```

Scrollup Character and Function

Qedit offers many ways to move around in a file while keeping the number of keystrokes to a minimum. For example, in full-screen mode you can enter the one of the following characters on the homeline or use one of the function keys.

Character	Function Key	Operation
+	F2	Scroll forward **Roll** number of lines
-	<none>	Scroll backward **Roll** number of lines
<	F5	Scroll backward one screenful
>	F6	Scroll forward one screenful

Enter the appropriate character on the homeline and hit ENTER or F7. Notice there's no equivalent function key for the minus sign "-". You had to bring the cursor to the homeline, enter the minus sign and hit F7 or ENTER.

You can now enter a minus sign in one (or both) copy/paste columns in full-screen mode. A single character scrolls up the number of lines defined in the **Set Visual Roll** option. Enter 2 minus signs to scroll up twice the number of lines and so on.

The minus sign is the default scrollup character. You can change it to another character that you may find easier to type. It must be a printable character and must not be a valid copy/paste code. Valid codes are: A, B, C, D, F, G, H, J, M, P, R, V, Z, ? and !. Use **Set Visual Scrollup "c"** to change the character.

```
/V Visual
Set VIsual Label ON Save OFF CLEar ON Renum ON Above 0 BElow 19 ROll 6
Set VIsual TAE OFF MSG ON HOme ON Widen 80 Carry 1  {Reflect=920W
Col=80}
Set VIsual BELL <null> ESC <null> FIeld "~" WOrdwrap OFF HIdetags OFF
Set VIsual HALfbright OFF HOStprompt OFF Inschar OFF Tab <null>
Set VIsual Update OFF CutCurrent ON SCReen OFF SO <null> SI <null>
Set Visual IgnoreLF OFF EDitonopen ON BUF 12000 Marginfixed OFF
Set Visual XX <none> Scrollup "-"                    ~-line: 1.59

/S Visual Scrollup "+"
/V Visual
Set VIsual Label ON Save OFF CLEar ON Renum ON Above 0 BElow 19 ROll 6
Set VIsual TAE OFF MSG ON HOme ON Widen 80 Carry 1  {Reflect=920W
Col=80}
Set VIsual BELL <null> ESC <null> FIeld "~" WOrdwrap OFF HIdetags OFF
Set VIsual HALfbright OFF HOStprompt OFF Inschar OFF Tab <null>
Set VIsual Update OFF CutCurrent ON SCReen OFF SO <null> SI <null>
Set Visual IgnoreLF OFF EDitonopen ON BUF 12000 Marginfixed OFF
Set Visual XX <none> Scrollup "+"                    ~-line: 1.59
```

You can enter **Set Visual Scrollup ""** to reset it back to the default character.

Merge Lines Horizontally

The Merge command replaces existing lines with the corresponding line from the external file. The Justified option appends the corresponding line from the external file. Text is appended immediately after the last non-blank character if **Set Work Trailingspaces** is disabled. If Trailinspaces is enabled, text is appended

immediately after the last significant trailing space. If the resulting merged line is too long for the current length, the merged line is truncated. Let's say the current workfile contains:

```
abc
def
ghj
```

and the external file contains:

```
1111
2222
3333
```

A MergeJ would result in:

```
abc1111
def2222
ghj3333
```

If the maximum length was 5, the resulting file would be:

```
/mergej myfile
    1     11111
Warning:  Result line will be too long.  Truncating merged text.
    2     22222
Warning:  Result line will be too long.  Truncating merged text.
    3     33333
Warning:  Result line will be too long.  Truncating merged text.
3 lines merged
/l all
    1     abc11
    2     def22
    3     ghj33
```

Preserving Trailing Spaces

By default, Qedit preserves trailing spaces on all lines in a variable-length file. Set Work Trailingspaces ON requests that Qedit preserves trailing spaces and make them significant characters. The option also allows creation of odd-length lines.

Once enabled, all workfiles created or opened from that point will have trailing spaces preserved. To check the current status, do:

```
/Verify Work       { Checks global setting }
Set WOrk Jumbo ON Block 8 Labels OFF Temp ON Size 3200 Random ON
Set WOrk TRailingspaces ON
/Verify Keep       { Checks current workfile }
Set Keep Ascii OFF Cctl OFF COde 0 Lab 0 Num OFF Var ON Checktimestamp
ON
Set Keep COBfree ON NAme /home/user1/afile.txt
Set Keep LF ON Saved modification timestamp 2003/04/30  13:23:17
Trailing spaces preserved
```

The last line shows that trailing spaces are preserved in this workfile. If the option is disabled, that line reads `Trailing spaces trimmed`. Disabling the global setting with Set Work Trailing Off does not disable the option in the workfile. You have to clear the workfile after disabling it.

The Trailingspaces setting is stored in the workfile so it's recognized when the file is opened in the future. These workfiles may contain data specific to Trailingspaces. This may cause unexpected behavior if opened with versions prior to 5.4.11. Because trailing spaces are now treated as significant characters, Keep files created from these workfiles may be different from Keep files created with an older version.

Variable Substitution

When the new Varsub option is enabled (**Set Varsub On**), Qedit parses entered commands looking for variable names. If a variable name is found and currently exists, its value is substituted before the command is executed. If the variable does not exist, the variable name is left unchanged.

Warning: The trailing comments limitation is an incompatibility with older versions.

Qedit commands are added to the Redo stack before the substitution occurs i.e. with the variable name. So, if the variable value changes between the time the command is entered and the time it is retrieved from the stack, the results may be different. It's also important to note that commands related to Redo stack operations such as **Listredo**, **Do**, **Before** can not have trailing comments enclosed in curly braces anymore. The comments are not removed and likely cause a syntax error.

```
/listredo  { see which commands I have entered so far }
Bad option, expecting ;UNN ;ABS ;REL or ;OUT
/listredo
     1) t testisql
     2) l "$myvar"
     3) s varsub on
     4) l "$myvar"
     5) setvar myvar "qed"
     6) l "$myvar"
     7) LISTREDO  { SEE WHICH COMMANDS I HAVE ENTERED SO FAR }
```

Variable names are identified by a leading dollar sign "$". For example, `$HOME` is replaced with the current value of the HOME environment variable. Some Qedit commands such as List have an extensive series of $-options which, as their name implies, also start with a dollar sign. These options have precedence over environment variables. In other words, if a variable has the same name as a $-option, the substitution does not occur. The only workaround is to change the name of the variable to something that does not conflict.

If you wish to prevent variable substitution and have Qedit interpret the dollar sign at face value, insert a backslash immediately in front of it as in `\$HOME`.

The tilde is a special character with different meanings in Qedit. Among other things, it can be a string delimiter or a shortcut pointing to the most recent current line in full-screen mode. In HP-UX shells, it's also commonly used to designate the user's home directory.

Here's how Qedit handles the tilde character. If it's still part of the string delimiter list (**Verify Stringdelimiter**), it is used as such. If it's not part of the list and is entered by itself on a line, it's interpreted as the most recent current line of full-screen mode. If it's not part of the list, Set Varsub is enabled and is used anywhere else in a command, it's replaced with $HOME. The Varsub feature would then substitute the appropriate value. Set Varsub On automatically removes the tilde from the string delimiter list.

The option is disabled by default.

Temporary Files

Qedit uses temporary files to perform edit operations. These temporary files include the primary scratch file and the explicit and implicit Hold files. These files are assigned random names using the HP-UX `tempnam` routine. The scratch file is called `qscr.`*`xxxxxxxxx`*, the explicit Hold file is `qhold`*`xxxxxxxxx`* and the implicit Hold file is `qhold`*`xxxxxxxxx`*`.0` where "xxxxxxxxx" is a random string.

Qedit works with their fully-qualified absolute names which can not have more than 240 characters.

By default, they are created in the /var/tmp directory. If you wish to have the files created elsewhere, you can specify the new location in the TMPDIR environment variable.

```
TMPDIR=/home/user1/tmp
export TMPDIR
```

[5.4.11]

Invalid Line Numbers

When Qedit reads numbered files, it assumes that each line carries a valid number and that the numbers are in ascending sequence. If it encounters an invalid number, it displays the following message:

```
Error: line number out of sequence (001200) - renumbering the rest
```

The string in parentheses is the invalid value. You should make sure that it contains numeric digits only and that it's greater than the number on the previous line. In order to check and fix this, you might have to text the file in using the **Unnumbered** option.

[5.4.11]

Sorting Lines

The **Lsort** command, as its name implies, sorts a range of lines. The syntax is:

LSORT *range* [KEYS *keylist*]

LSORT *string range* [KEYS *keylist*]

(Q=no display)

(Default: by entire line)

The simplest Lsort command just specifies a range of lines to be sorted and no other parameters. This means to use the entire line as the key and sort the lines into ascending order, printing them once sorted.

To stop Lsort from printing the sorted lines, use LsortQ. The Lsort command can be abbreviated to "lso", "lsq" (quiet), "lst" (template) and "lsj" (justify). "ls" followed by a space executes the HP-UX shell command. "ls" followed by any other character is executed as a possible shell command.

Parameters

To sort by some other key fields in the lines (from one to four are supported) or to sort the lines in Descending Order, you need to specify the KEYS *keylist* parameters. The *keylist* consists of one to four keys separated by spaces or commas, with a key consisting of either a column range or a starting column and length:

column , length [DESC]

column / column [DESC]

Ascending Order is assumed by default, but you may specify DESC to sort this key in Descending Order.

Examples

```
/lsort all           {sort entire file}
/lsortq all          {sort without printing}
/lsort 10/33         {sort some lines only}
/lsort 30/last keys 10,5           {col 10 through 14}
/lsort zz keys 10/20               {col 10 through 20}
/lsort 20/last keys 1,10 20,5,desc  {two keys}
```

[5.4.10]

Cobol File Extension

Qedit looks at different file attributes to try and determine the type of file it's dealing with. Cobol source files can be identified by the file extension (.cob, .cbl, .CBL). A new file extension, `.pco`, has been added to the list. This extension is typically used for files that need to be processed by the Oracle pre-compiler.

[5.4.10]

Form Command

Displays information about a self-describing file created by programs such as Suprtool. These programs store information about the record layout such as field names, data types, length.

FORM [$LP | $LPA | $LPB] [*filename*]

(Default: *filename* = current Text file)

If *filename* is omitted and a workfile is currently active, Qedit uses the name of the Text file (see **Verify Keep**). An external filename can be specified. In this case, the name must be the name of the data file.

If the file is not self-describing, Qedit displays the following message:

```
Error:  File is not self-describing.
```

Self-describing files on Hp-UX have 2 components: the data file and the data description file. The name of the data description file is the name of the data file followed by the `.sd` extension. For example,

```
/home/user1/mydata.dat          { data }
/home/user1/mydata.dat.sd       { data description }
```

The Form output looks like this:

```
Self-describing information for /home/user1/mydata.dat
    File: /home/user1/mydata.dat     (SD Version B.00.00)  Has
linefeeds
       Entry:                      Offset
          CHAR-FIELD           X5      1  <<Sort# 1 >>
          INT-FIELD            I1      6
          DBL-FIELD            I2      8
          PACKED-FIELD         P12    12
          PACKED*-FIELD        P12    18
          QUAD-FIELD           I4     24
          ID-FIELD             I1     32
          LOGICAL-FIELD        K1     34
          DBLLOG-FIELD         K2     36
          ZONED-FIELD          Z5     40
    Entry Length: 80  Blocking: 1
```

LP Listing

$lp, $lpa and $lpb send output to a device associated with an environment variable of the same name. For example, to print to the device called Laser with the $lpa option, you must set the |4LPA| environment variable to Laser, as in |2export lpa=laser|. If the LP environment variable is not set, Qedit will attempt to send the output to the default system printer. But if the LPA or LPB environment variables are not set with a valid device name, an error will occur.

[5.4.10]

Colcopy Command [COL]

Copies one or more columns to a different location on the same line.

COLCOPY *source* [*/source2*] *destination1* [*/destination2*] [*rangelist*]

(Q=no display, J=verify, T=CobX Tag)

(Default: *rangelist* = *)

Colcopy copies text in columns specified by *source1* and *source2* to the destination columns specified by *destination1* and *destination2* in the lines of *rangelist*. Even though Colcopy can modify multiple lines using a *rangelist*, it really operates on one line at a time. You can not copy columns from one line to another.

Source and destination columns always represent the original location. All changes are based on that assumption.

If *source1* only is specified, Qedit copies just that column (length of 1). If *destination1* only is specified, the source columns are inserted at that location. If you wish to replace a single column, enter a destination range where *destination1* and *Destination2* are the same e.g. `Colcopy 1 10/10`.

```
/list 1
   1     abcdefghijklmnopqrstuvwxyz
/colcopy 1 10          { insert column 1 at column 10 }
   1     abcdefghiajklmnopqrstuvwxyz
1 line changed
/colcopy 1/5 10        { insert columns 1/5 at column 10 }
   1     abcdefghiabcdejklmnopqrstuvwxyz
1 line changed
```

If *destination1* and *destination2* are specified, text in these columns is replaced by the source text. If the source text is narrower or wider, the line is shortened or expanded as needed.

```
/colcopy 1 10/15     { copy column 1 to columns 10/15 }
   1     abcdefghiapqrstuvwxyz
1 line changed
/colcopy 1/5 10/11   { copy columns 1/5 to 10/11. Line expands. }
   1     abcdefghiabcdelmnopqrstuvwxyz
1 line changed
/colcopy 1/5 10/20   { copy columns 1/5 to 10/20. Line shortens. }
   1     abcdefghiabcdeuvwxyz
1 line changed
```

Trailing Spaces

Trailing spaces on the line are not significant. This means that a line can expand until a non-space character reaches the current right margin (**Set Right**). However, trailing spaces from the source text are significant and are copied in the operation. If the line can not be expanded further, Qedit displays a warning message and allows the user to modify it.

```
/list 2
   2     abcd     efghiabcdeuvwxyz
/colcopy 1/8 20        { insert columns 1/8 at 20 }
   1     abcd     efghiabcdeabcd    uvwxyz
1 line changed
/Set right 30
/colcopy 1/5 30        { insert columns 1/5 at 30 }

Warning: Source columns could not be inserted. Please modify. (Warning
2)
   1     abcd     efghiabcdeabcd    uvwxyz
1 line modified
```

Overlapping Columns

When source and destination columns do not overlap, the results are straightforward. If source and destination columns overlap partially or completely, the results might not be as expected. Keep in mind that:

- source and destination columns are always based on the original line
- the destination columns are removed
- the source columns are put in their place

Approving Each Changed Line

Use COLJ to give yourself approval over each change before it is updated. With COLJ, Qedit displays the line as it would be and asks you for a Yes, No, or Modify answer.

CobX Tags

Cobol tags are short strings stored in columns 73 to 80 of CobX source files. The Cobol tag value is defined using the Set X command. Once enabled, updated lines and added lines are automatically updated with the tag. They can also be modified manually with custom tag values.

In its regular form, the Colcopy command affects only the text area in columns 7 to 72. If you wish to make changes to Cobol tags, use the `T` suffix. You can think of it as the `Tag` option. This option operates only on the tag area itself, columns 73 to 80.

```
/ColT 73/74 79/80 all      { copies content of columns 73 and 74 }
                           { into columns 79/80                  }
/ColT 73/74 75 all         { inserts content of columns 73 and 74     }
                           { in column 75. Columns 76-80 are shifted. }
```

To to this, the `Tag` option temporarily changes the margins to `(73/80)`. Qedit displays a warning every time this option is used. Because the margin values have changed, explicit column range in the source and destination columns can only be between 73 and 80.

```
/ColT 23/24 79/80 all
Warning:  ColcopyT: editing the Cobol tag area only (73-80).
Error:  The Sourcestart column (23) is not between 73 and 80

/ColT 73/74 79/80 10
Warning:  ColcopyT: editing the Cobol tag area only (73-80).
   10      ME0307ME
1  line changed
```

Because the margins have been changed, Qedit displays text in the tag area only except when the Justify option is used. In this case, Qedit prompts for confirmation before making the change. It would be hard to determine if a line needs to be changed based only on the tag value. So, when the Justify option is used, Qedit displays the complete line. The user has the option to accept the changes, reject the

changes or manually modify the line. If the user chooses to modify the line, only the tag is displayed.

[5.4.10]

Colmove Command [COLM]

Moves one or more columns to a different location on the same line.

COLMOVE *source* [*/source2*] *destination1* [*/destination2*] [*rangelist*]

(Q=no display, J=verify, T=CobX Tag)

(Default: *rangelist* = *)

Colmove moves text in columns specified by *source1* and *source2* to the destination columns specified by *destination1* and *destination2* in the lines of *rangelist*. The source columns are removed from their original location. Even though Colmove can modify multiple lines using a *rangelist*, it really operates on one line at a time.

You can not move columns from one line to another. Source and destination columns always represent the original location. All changes are based on that assumption.

If *source1* only is specified, Qedit moves just that column (length of 1). If *destination1* only is specified, the source columns are inserted at that location. If you wish to replace a single column, enter a destination range where *destination1* and *Destination2* are the same e.g. `Colcopy 1 10/10`. A move means the original columns are removed and the line is shifted left. Then the source text is inserted at the destination.

```
/list 1
   1     abcdefghijklmnopqrstuvwxyz
/colmove 1 10          { move column 1 to column 10 }
   1     bcdefghiajklmnopqrstuvwxyz
1 line changed
/colmove 1/5 10        { move columns 1/5 to column 10 }
   1     fghiabcdejklmnopqrstuvwxyz
1 line changed
```

If *destination1* and *destination2* are specified, text in these columns is replaced by the source text. If the source text is narrower or wider, the line is shortened or expanded as needed.

```
/colmove 1 10/15     { move column 1 to columns 10/15 }
   1     bcdefghiapqrstuvwxyz
1 line changed
/colmove 1/5 10/11   { move columns 1/5 to 10/11 }
   1     fghiabcdelmnopqrstuvwxyz
1 line changed
/colmove 1/5 10/20   { move columns 1/5 to 10/20 }
   1     fghiabcdeuvwxyz
1 line changed
```

Trailing Spaces

Trailing spaces on the line are not significant. This means that a line can expand until a non-space character reaches the current right margin (**Set Right**). However, trailing spaces from the source text are significant and are moved in the operation.

```
/list 2
   2     abcd     efghiabcdeuvwxyz
/colmove 1/8 20        { move columns 1/8 to 20 }
   1      efghiabcdeabcd    uvwxyz
1 line changed
```

Overlapping Columns

When source and destination columns do not overlap, the results are straightforward. If source and destination columns overlap partially or completely, the results might not be as expected. Keep in mind that:

- source and destination columns are always based on the original line
- the source columns are removed
- the destination columns are removed
- the source columns are put in their place

Here is an example:

```
/list 1
   1     abcdefghijklmnopqrstuvwxyz
/colm 6/20 15
   1     abcdefghijklmnopqrstuvwxyz
1 line changed
```

Apparently, nothing has changed but, in fact, something did happen to the line. Qedit removed the source columns "fghijklmnopqrst" and tried to insert the original text where column 15 used to be. Column 15 was part of the area that has been removed so Qedit inserts the text where it should have been i.e. between "e" and "u". So, it's putting the original text back where it was.

Moving Passed the Right Margin

Destination columns can exceed the current right margin. In this case, Qedit assumes the columns should be moved to the end of the line. Effectively, the source columns are inserted in the rightmost columns of the line. The destination columns do not have to be a precise value. They just need to be larger than the current right margin. If the right margin is currently set at 80, the following commands yield the same results.

```
/v right
Set Right 50
/lt2
         ....+....10...+....20...+....30...+....40...+....5
   2     abcdefghijklmnopqrstuvwxyz
/colm 1/5 51
   2     fghijklmnopqrstuvwxyz                          abcde
1 line changed
/colm 1/5 88/90
   2     fghijklmnopqrstuvwxyz                          abcde
1 line changed
```

Approving Each Changed Line

Use COLMJ to give yourself approval over each change before it is updated. With COLMJ, Qedit displays the line as it would be and asks you for a Yes, No, or Modify answer.

CobX Tags

Cobol tags are short strings stored in columns 73 to 80 of CobX source files. The Cobol tag value is defined using the Set X command. Once enabled, updated lines and added lines are automatically updated with the tag. They can also be modified manually with custom tag values.

In its regular form, the Colmove command affects only the text area in columns 7 to 72.If you wish to make changes to Cobol tags, use the `T` suffix. You can think of it as the `Tag` option. This option operates only on the tag area itself, columns 73 to 80.

```
/ColmT 73/74 79/80 all     { copies content of columns 73 and 74 }
                           { into columns 79/80                  }
/ColmT 73/74 75 all        { inserts content of columns 73 and 74     }
                           { in column 75. Columns 76-80 are shifted. }
```

To to this, the `Tag` option temporarily changes the margins to `(73/80)`. Qedit displays a warning every time this option is used. Because the margin values have changed, explicit column range in the source and destination columns can only be between 73 and 80.

```
/ColmoveT 23/24 79/80 all
Warning:  ColcopyT: editing the Cobol tag area only (73-80).
Error:  The Sourcestart column (23) is not between 73 and 80

/ColmoveT 73/74 79/80 10
Warning:  ColcopyT: editing the Cobol tag area only (73-80).
   10      ME0307ME
1  line changed
```

Because the margins have been changed, Qedit displays text in the tag area only except when the Justify option is used. In this case, Qedit prompts for confirmation before making the change. It would be hard to determine if a line needs to be changed based only on the tag value. So, when the Justify option is used, Qedit displays the complete line. The user has the option to accept the changes, reject the changes or manually modify the line. If the user chooses to modify the line, only the tag is displayed.

[5.4.10]

Excluding Lines From Visual Mode Display

The XX indicators are used to mark lines that you do not want displayed in full-screen mode. Once marked, the block of text is replaced with a single line.

```
--- Excluded Area ---  10/34.5
```

This line shows the line numbers which are currently excluded. An excluded area setting is saved in the workfile so it's preserved across **Open**/**Shut** commands. To reset the excluded area and see the original lines again, type `.xx` on the Homeline and press Enter or F7.

The excluded area can also be defined using **Set Visual XX**. The current excluded area is displayed on the **Verify Visual** output.

Restrictions

The `Excluded Area` line must not be removed, altered or used in any way. This also means that you can not enter any indicators in the cut-and-paste area. If you wish to paste lines before or after the excluded area, you should use the appropriate cut-and-paste indicators on the line that immediate precedes or follows the `Excluded Area` line.

An excluded area can not be included in any other block operation such as ZZ, CC, MM or other XX.

If any of these rules are broken, Qedit displays an appropriate error message.

Set Visual XX

Set Visual XX [*startline* **[/** *endline* **]]**

(Default=reset)

Set Visual XX defines the lines that should be excluded from the full-screen mode display. Excluded lines are replaced by a single line.

```
--- Excluded Area ---  10/34.5
```

This line shows the line numbers which are currently excluded. If no parameters are specified, the current excluded area is reset. An excluded area must have a start and an end line. If only *startline* is specified, the excluded area is incomplete. An appropriate message is going to be displayed on the status line next time the user goes into full-screen mode.

To complete the excluded area, enter another Set Visual XX command with another line number. This number is going to be used as the *endline*. Of course, you can specify both *startline* and *endline* on a single command.

```
/Set Visual XX 5      { Sets the start line. XX incomplete. }
/Set Visual XX 10     { Sets the end line. XX=5/10          }
/Set Visual XX 5/10   { Sets XX to 5/10 }
```

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 5.4

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

- The server and client views of a file were sometimes different after pasting some lines in a Cobol source file. The problem could lead to file corruption, strange behavior and even client aborts. [5.4.10]
- The server notifies the Qedit for Windows client whenever it runs out of line numbers (Error: Too High). Qedit for Windows displays an appropriate message and the file is still opened but it may be missing some lines. [5.4.10]
- Running Qedit with Blockemulation mode enabled does not work with QCTerm 1.02 or older. It works on version 2.0 and higher. QCTerm is a free terminal emulator from AICS Research. [5.4.11]
- Full-screen did not always change the display width automatically in Qcterm. [5.4.11]
- The Qedit for Windows client sometimes aborted with a linecount discrepancy error after a paste operation. This was caused by a problem in the server. [5.4.11]
- Interrupting a listing with Control-C (interrupt signal equivalent to Control-Y on MPE) after exiting from Screen mode does not cause an abort anymore. [5.4.12]
- Trying to copy/move a block of lines where the start or end of the block is on a newly inserted line (relative number not assigned yet) in full-screen mode returned an error message: Copy/paste functions not allowed on Excluded Area line. [5.4.12]
- Inserting new lines in full-screen mode with the Excluded Area line displayed and causing a background renumbering operation sometimes lead to file corruption (lines inserted in the wrong place, duplicate line numbers). [5.4.12]
- Interrupting a listing with Control-C (interrupt signal equivalent to Control-Y on MPE) after exiting from Screen mode does not cause an abort anymore. [5.4.12]
- **Colcopy** and **Colmove** commands inserted invalid characters when the source columns exceeded the current line length. [5.4.13]

- An **Undo** command sometimes reported an error when the **TrailingSpaces** option was enabled. [5.4.13]