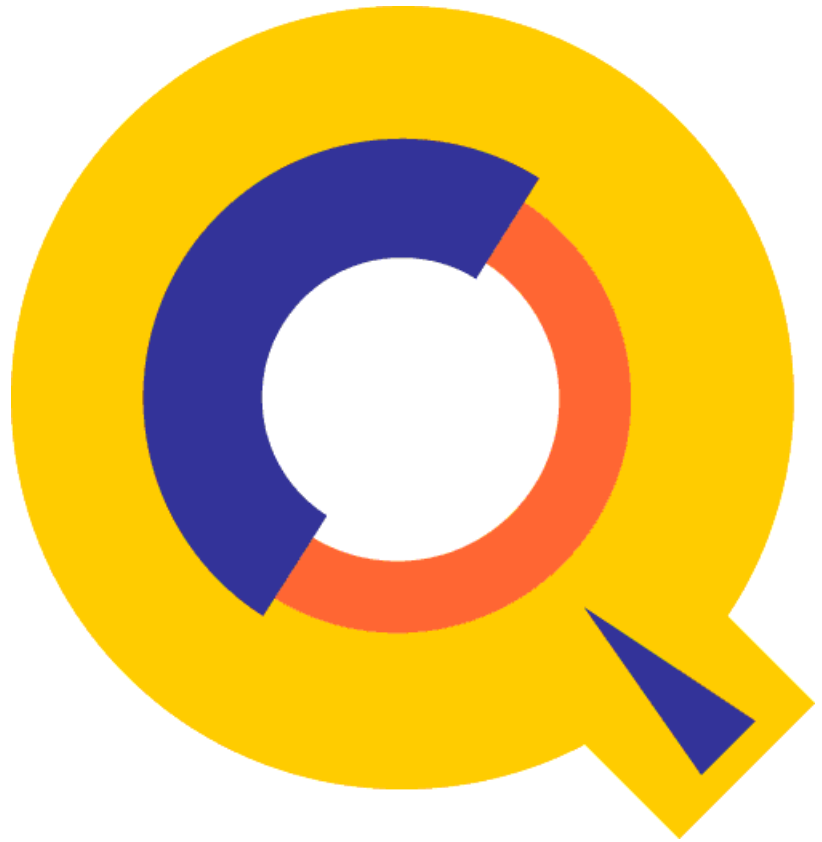

Qedit 5.4 for HP e3000

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

E-mail: support@robelle.com

Web: www.robelle.com

Contents

Introducing Qedit 5.4	1
Introduction.....	1
Qedit for Windows.....	1
Highlights In Version 5.4.....	1
Line Mode	1
Full-screen Mode.....	2
Compatibility	2
Unresolved Externals on MPE/iX 4.0	2
CPU Serial Number (HPSUSAN)	3
Compatibility Mode vs. Native Mode	3
Known Problems	3
Documentation.....	4
User Manual	4
Change Notice	4
Printed Documentation.....	4
Online Documentation in HTML Help Format	4
CD or Web Download	4
Installation	5
Upgrade Instructions.....	5
Installation Assistance	5
Important Note About Passwords	5
STREAMX Users.....	6
Combined Qedit/Suprtool Tapes	6
Step 1: Install Qedit	6
Restore All Robelle Files.....	6
Create/Update Robelle Account	7
File Names.....	7
Install Program Files.....	7
Step 2: Install QLIB and Bonus Programs.....	8
Building the Spell Dictionary	8
Step 3: Install NM Compiler Interface (only XL).....	9
Installing the Interface	9
Compiling Instructions	9
Step 4: Install CM Compiler Interface (optional)	9
Choosing a CM Installation Method.....	10
Integrating CM Compiler Changes.....	10
Isolating CM Compiler Changes	10
Step 5: Purge Installation Files (optional).....	11
Enhancements in Version 5.4	13
Introduction.....	13
String Range	14

Lines, Strings and Ranges.....	14
Delete With Confirmation.....	16
Most Recent Visual Line	16
Right Margin Control.....	16
Type 1234 Terminals and Emulators	17
List Nearest Line.....	18
Qedit for Windows	19
Introduction.....	19
Server Process.....	19
Logon Sequence	19
Firewall Protocol and the Server	20
QEDSERVMODE JCW	20
Bugs Fixed Since Version 5.3	21
Introduction.....	21
Search for String on Visual.....	21
Host File Corruption.....	21
Temporary Window Not Used.....	21
Clearing Display Memory	22
Find and FindUp Aborts.....	22
Server Loop	22
Comment Lines in Command Files	22

Introducing Qedit 5.4

Introduction

Qedit is our fast and efficient full-screen editor and development home base for HP e3000 systems (MPE/iX and MPE V). 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 HP-UX, 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

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 tilde character "~" can be used in line-mode to reference the last line accessed in full-screen mode (**Visual**). By default, it's interpreted as a string delimiter. 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.
- **Set List Nearest** option determines whether a **List** command on a non-existent line number displays a warning or the nearest line.
- Lines starting with a pound sign "#" in a command file are now recognized as comments. As such, they are not interpreted and I/O redirection characters "<" and ">" do not cause errors.

Full-screen Mode

- 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.
- 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 Visual Clear Off was not working when running on Reflection for Macintosh.
- 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.

Compatibility

Unresolved Externals on MPE/iX 4.0

If you try to run Qedit on MPE/iX 4.0, you might get the following errors:

```
UNRESOLVED EXTERNALS: _thd_errno (LDRERR 512)
UNRESOLVED EXTERNALS: _thread_set_error (LDRERR 512)
UNRESOLVED EXTERNALS: thd_lock_lang_mutex (LDRERR 512)
UNRESOLVED EXTERNALS: thd_unlock_lang_mutex (LDRERR 512)
```

These messages indicate some routines missing from the C library. If you are not planning to upgrade to a more recent version of MPE/iX, you should contact Robelle technical support. We can provide you with an additional file that will fix these problems.

CPU Serial Number (HPSUSAN)

Qedit runs only on CPUs whose serial numbers (the "showvar hpsusan" number on MPE/iX) have been encoded. If it fails to run and you see an "invalid HPSUSAN" error message, call Robelle for assistance.

Compatibility Mode vs. Native Mode

Throughout this change notice and in all the Qedit documentation, there are references to compatibility mode and native mode. If you are on a Classic HP e3000 (that is, not on MPE/iX), you can use the compatibility-mode version of Qedit. Our installation jobs will automatically install the correct version of Qedit for your machine.

Qedit works on both MPE/iX and MPE XL. Any reference to MPE/iX in our documentation can also be considered a reference to MPE XL, and vice versa.

Known Problems

The underscore character can be used as a string delimiter and as a valid character in a Posix filename. If you enter a filename containing an underscore in the INFO= string parameter, Qedit interprets the underscore as a string delimiter. For example, if you enter:

```
run qedit.pub.robelle;info='-c "t my_history;vi"'
```

This command should start up Qedit, have it open the Posix file called `my_history` and go into full-screen mode automatically. However, since the underscore is seen as a string delimiter, whatever follows is basically ignored during command processing. Thus, the file is opened but Qedit stays at the line-mode prompt.

We hope to have a permanent fix in a future release. In the meantime, the workaround is to explicitly identify the underscore as being part of the filename. You can do this in different ways. Here are two examples.

```
run qedit.pub.robelle;info='-c t $file \my_history\;vi''
```

The `$file` keyword and the backslash characters are used to clearly identify the filename. The other approach is to remove the underscore from the list of valid delimiters using the `Set StringDelimiters` command.

```
run qedit.pub.robelle;info='set stringd posix;-c "t my_history;vi''
```

The `$file` approach is simpler as it only affects that particular command. The `Set StringDelimiter` approach should be used with caution as it might affect other commands during the edit session.

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

Upgrade Instructions

If you are a *new Qedit user*, please follow the installation instructions in the *Qedit User Manual*. The following installation instructions assume that you are an existing Qedit user who is upgrading to version 5.4.

To install this upgrade of Qedit, follow these steps:

1. Install Qedit.
2. Install the QLIB and/or Bonus programs.
3. Fix the NM compilers. (only XL)
4. Fix the MPEV compilers. (optional)
5. Purge installation files. (optional)

Installation should take less than 20 minutes. No one can use Qedit while step 1 is in progress. Bonus and QLIB programs cannot be used while step 2 is in progress.

Installation Assistance

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

Important Note About Passwords

None of the jobs we supply have passwords in them. Before streaming a job, you might have to add your system's passwords to the first line. Users of MPE XL version 3.0 and later do not have to do this because the operating system prompts you for missing passwords. The same is true for MPEV users who have security software that inserts passwords. Most MPEV users have to edit the jobs. For example, if the system manager logon password is Qwerty, you would do the following:

```
:editor
HP32201A.07.22 EDIT/3000
/text robelle.job.robelle
FILE UNNUMBERED
/modify 1
MODIFY      1
!job robelle,manager.sys,pub;hipri
              i/qwerty
!job robelle,manager.sys/qwerty,pub;hipri

/keep robtemp
/exit
END OF SUBSYSTEM
:stream robtemp
:purge robtemp
```

STREAMX Users

Users of STREAMX, a part of SECURITY/3000 from VESOFTE Inc., must set a Job Control Word before streaming jobs. This step prevents STREAMX from generating an error if the Robelle account does not yet exist. For example,

```
:setjcw streamxtemponest 1
:stream robelle.job.robelle
```

Combined Qedit/Suprtool Tapes

If the tape you received has more than one product on it (e.g., Qedit and Suprtool), you can take a shortcut instead of following the complete instructions for both products.

Steps that are common to both installations need only be done once. That is, there is no need to restore twice, or setup the Robelle account twice, or install the Qlib and Bonus programs twice, or clean up unneeded files twice.

Note that if you do *not* use this shortcut it is not a problem; it will just take you a few minutes longer, and you will have to mount the restore tape a second time.

Say you install Qedit first, followed by Suprtool:

1. Do all of the Qedit steps, except for the final *Purge Installation Files* step.
2. Then do the Suprtool installation starting with the *:stream Install.Suprjob* step. Skip the initial *Restore* and *Robelle Account* steps, and skip the *Qlib/Bonus* step. Do the final *Purge Installation Files* step.

Step 1: Install Qedit

This step requires three separate procedures: restoring all Robelle files from tape to disc, building (or upgrading) the Robelle account with the job stream that we provide, and streaming the installation job stream. All these procedures can easily be accomplished if you log on as Manager.Sys.

Restore All Robelle Files

First, you have to restore all the files from tape.

```

:hello manager.sys           {log on as system manager}
:file rtape;dev=tape         {mount Robelle tape}
:restore *rtape; @.@.robelle; create  {=reply on the console}

```

Check the :Restore listing for files that were busy and could not be restored. There are a number of reasons why a file could have been busy: someone may have been using it or backing it up, it may have been held as a suspended son process (menu system), or it may have been allocated. If files were busy, chase away any users, deallocate any busy programs, then try the :Restore command again.

Create/Update Robelle Account

You *must* build (or upgrade) the Robelle account with the job stream that we provide. This job removes the passwords from the Robelle account and from the user Mgr.Robelle. You need to know the existing passwords so that you can put them back after the installation.

```

:listacct robelle;pass       {look for PASSWORD: }
:listuser mgr.robelle;pass   {look for PASSWORD: }

```

Once you have written down the passwords, stay logged on as Manager.Sys and stream the job that builds and updates the Robelle account.

```

:stream robelle.job.robelle

```

This job stream will send you a message when it is finished.

Stay logged on as Manager.Sys and put a password on the Robelle account. If you are upgrading your Robelle account, use the old password.

```

:altacct robelle;pass=hardpass {hard to guess}

```

Please note that during installation, we add OP capability to the Robelle account. Once Qedit has been installed, you can remove OP capability.

File Names

The following Qedit program files are restored to the Pub group of the Robelle account:

Program	Purpose
qeditnm	NM Qedit program
qeditcm, qeditpm	CM Qedit programs
qcompxl	NM compiler interface
qedify, qcompusl	CM compiler interface
qloadxl	NM Qedit options
ederrnm, ederrcm	:Editerror support
qmap	reformat MPEV PMAP

Install Program Files

Our installation job puts the proper Qedit files into the Pub group of the Robelle account. To make sure no one uses these programs during the installation, send a warning. Then stream the job:

```
:hello mgr.robelle
:warn @:please exit from Qedit NOW!

:stream install.qeditjob          {supply passwords}
```

Check the installation job \$stdlist. If anyone was using Qedit or attempting to back it up during the installation, the job will fail. If the job failed, chase away any users, ensure that a backup is not in progress, then stream the installation job again.

The installation job renames your current versions of Qedit to the PubOld group of the Robelle account. If you need to move these versions back into production, use the Previous.Qeditjob job stream.

Once you have installed Qedit, users can now access it by entering

```
:run Qedit.pub.robelle
```

Step 2: Install QLIB and Bonus Programs

Qedit comes with an array of contributed software in the QLIB library (in the QLIB groups of the Robelle account). QLIB programs may be used on any CPU and may be given away freely. Your Robelle license may also entitle you to receive our Bonus programs: Compare/iX, HowMessy, Select, Spell, and Xpedit. These programs reside in the Pub group of the Robelle account. Bonus programs can be used only on authorized machines, and you may not distribute them to anyone.

If you received Bonus programs with this version of Qedit, use the job stream called Bonus.Job.Robelle to install both the QLIB and Bonus programs. If you did not receive Bonus programs, use the job stream Qlib.Job.Robelle to install the QLIB programs.

```
:hello mgr.robelle
:stream bonus.job.robelle          {or Qlib.Job.Robelle}
```

If you skip this step, you may end up running old versions of these programs.

Building the Spell Dictionary

Before you can use the Spell Bonus program or the Words command in Qedit, you have to build the main dictionary for Spell with an additional installation job. This job stream can take 30 to 60 minutes; you do not have to wait for it to finish before going to the next step.

By default, the Spell program uses American spellings. To use British spellings, set the SpellAmerican JCW to False in the installation job.

```
:run qedit.pub.robelle          {or use :Editor}
/text dictmain.spjob
/modify "setjcw SpellAmerican"  {for British spelling,  }
/keep robtemp                   {  change True to False}
/exit
:stream robtemp
:purge robtemp
```

For more information, see the *Spell User Manual*.

Step 3: Install NM Compiler Interface (only XL)

If you don't need to compile source files that are saved as permanent Qedit workfiles (file code=111), then this step is optional. It is also optional if you have not installed a new version of MPE since your previous Qedit installation.

If you use any NM compilers on MPE/iX, you will want to install the new NM compiler interface. If you still use the CM compilers as well, you will also want to install that compiler interface (described in step 4 below).

Installing the Interface

Assuming that you have restored the Robelle files on your system and that you have not done this step before, stream Savecmdf.Qeditjob to make backup copies of the existing HP command files in the Stdcmd.Sys group. You need to know the Manager.Sys password.

```
:stream savecmdf.qeditjob.robelle {wait for it to finish}
:listf @.stdcmd.sys
```

Then stream Qcompxl.Qeditjob to make three (3) changes to the Pub.Sys compiler command files:

```
:stream qcompxl.qeditjob.robelle
```

This job makes the following changes to the compiler command files:

1. Change the run of the compiler program to
;xl ="qcompxl.pubnew.robelle, qcompxl.pub.robelle".
2. Add ";shr" to the File commands for the text and master files.
3. Change "setjcw jcw=fatal" to an invalid command so that errors in either Qedit or MPE/iX will abort the command file.

Compiling Instructions

Now you should be able to compile a Qedit file or a Keep file, either within Qedit or at the MPE/iX prompt, simply by invoking the normal command file.

```
:pasxl source {from MPE}
:run qedit.pub.robelle
/pasxl source {from Qedit}
```

Within Qedit, you can specify an asterisk (*) for the current workfile.

```
:run qedit.pub.robelle
/text source
/pasxl *
/exit
:pasxl qeditscr
```

Step 4: Install CM Compiler Interface (optional)

If you don't need to compile source files that are saved as permanent Qedit workfiles (file code=111), then this step is optional.

When installing Qedit on MPE/iX, you might have to install the NM compiler interface (described earlier) and the CM compiler interface (described here). For MPEV, however, you only have to worry about the original CM compilers.

After restoring the files from tape, you have to choose an installation method: integral or isolated.

Choosing a CM Installation Method

There are two methods of fixing the compilers on your system: *integral* into Pub.Sys or *isolated* in the Robelle account. The isolated method is best for new or trial-period users because it is faster to install and easier to remove later. You can start with the isolated method and easily convert to the integral method later.

To determine if you currently use the isolated or integral method, run Qedit and enter

```
/verify whichcomp
```

If the displayed line ends with "IN ROBELLE," you are using the isolated method. If it ends in "IN SYS," you are using the integral method.

Integral Method.

The compilers in Pub.Sys are "fixed" directly, and a segment is added to the system SL. The standard :Cobol and :Pascal commands of MPE then compile Qedit workfiles. :Sysdump is used to make a new cold load tape for the next time you need to boot the system.

Isolated Method.

The MPE compilers are copied into the Q.Robelle group and the original compilers are left in the Pub.Sys group. No code is added to the system SL. Qedit workfiles can be compiled only within Qedit because the "Qedit" compilers are not in Pub.Sys.

Integrating CM Compiler Changes

The job stream Qeditj1.Qeditjob.Robelle makes *integral* compiler fixes. Use it either to update the fixes to a new version or to re-install the fixes after an MPE update from HP. Before you start, you should :Store @.Pub.Sys and save it on tape because this job modifies the compiler program files in Pub.Sys. You will need another small tape for a new cold load tape.

Ensure that no one compiles during the installation. Stop all jobs and send an operator warning. Log on as Manager.Sys, modify the first line of the Qeditj1 job to include the passwords, then :Stream the job.

```
:stream qeditj1.qeditjob.robelle
```

This job runs Qedify.Pub.Robelle, which updates the compilers in Pub.Sys and installs nonprivileged hooks into them so that the compilers can access Qedit workfiles.

The job adds a code segment from Qcompusl.Pub.Robelle into Sl.Pub.Sys and then requests a tape mount for a new cold load tape.

Mount a write-enabled tape and :Reply. Save this tape and use it for future cold loads. The CM compiler interface is now installed. Compiling done inside or outside Qedit will work on either Qedit workfiles or on regular Keep files.

Isolating CM Compiler Changes

The job stream Qeditj1a.Qeditjob.Robelle fixes the compilers, but it isolates the changes in the Q.Robelle group. Ensure that no one compiles during the installation.

Stop all jobs and send an operator warning. Modify the first line of Qeditj1a to include the Robelle password, then stream the job.

```
:stream qeditj1a.qeditjob.robelle
```

The job uses Qedify.Pub.Robelle to copy the compilers from Pub.Sys to the Q.Robelle group, then it "fixes" them so that they can read Qedit files. The original compilers in the Pub.Sys group are not modified. Once this step is finished, you can use the CM compilers on Qedit files from within Qedit, but not at the MPE prompt.

Step 5: Purge Installation Files (optional)

The easiest way to clean up after an installation is to stream the Cleanup.Purgejob.Robelle job. This job saves you disc space by purging the installation files. It does not purge the files that are necessary to run the software.

```
:stream cleanup.purgejob.robelle
```


Enhancements in Version 5.4

Introduction

With every release we provide Qedit users with new features. Major enhancements since the last major release include:

- 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. [5.3.14]
- 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]
- 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]
- **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]
- 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]

The following sections describe the new enhancements to Qedit 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.

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]

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 let's 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 jcws to their default state, forcing Qedit to re-identify the terminal. However, for hpterm, the RCRTMODEL and RCRTWIDTH jcws 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]

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 parts: the client and the 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

By default, the Qedit server uses the MPE/iX Remote Process Management (RPM) service to start each server session. With RPM, the client sends a program name and an Info= string along with logon information to the RPM listening process. The RPM listening process then validates the user logon and starts the specified program with the Info= string. The server process is seen as an MPE/iX session, but the process is not included in your user limit. The Qedit server software is not available for MPEV.

Logon Sequence

When you use an MPE/iX connection to open a file in Qedit for Windows, this is what happens:

1. A new host session is started, and the user name and passwords are validated (this includes additional logon processing by products such as SECURITY/3000).
2. The Qedit client asks RPM on the host to run the Ci.Pub.Sys program with an Info string. For example,

```
qedit.pub.robelle "-d<ip-address>"
```

The "<ip-address>" is the IP address of the Qedit client.

3. The Qedit process starts, and it verifies that you are authorized to use Qedit in server mode.

4. Any system-wide Qeditmgr files are always processed. Processing of account and group Qeditmgr files is optional and must be enabled in the client.
5. The requested file is opened.
6. A response is sent to the client.

It is important to note that logon UDCs are not executed as part of this process. Any environment variables or file equations that are set up as part of a logon UDC will not be applied in the Qedit server process.

Firewall Protocol and the Server

The default protocol is usually sufficient at most sites. However, some customers use firewall devices to further restrict access to their HP e3000. The default protocol might not work at some of these sites. For this reason, the server offers a Firewall protocol.

In order to use this protocol, a Qedit listener has to be running at all times. You should add this job to your system startup procedure:

```
:stream qserver.qeditjob.robelle
```

If the job is not running, all new connection requests are rejected. Currently, the only way to stop the job is to abort it.

With this setup, the server runs as a listener similar to a UNIX daemon. It waits for incoming connection requests on a specific port number. Like the HP-UX server, the default port number is 7395. If you want to use a different port, you have to modify the Qserver job stream.

Upon receiving a request, the listener passes the information down to RPM, which then takes over.

QEDSERVMODE JCW

The Qedit server process always executes the commands in the system-wide Qeditmgr files (e.g., Qeditmgr.Pub.Sys). Processing of the account and group Qeditmgr files can be enabled in the client. To indicate that you are using the Qedit server in either the server or the nonserver mode, which are not exactly the same, we provide a JCW called QEDSERVMODE. QEDSERVMODE is set to zero (0) if you are in nonserver mode, and nonzero if you are in server mode. If you are using the Qedit server, you should modify your Qeditmgr file so that few, if any, commands are executed in server mode. For example,

```
if Qedservmode = 0 then
  set extprog mpex.pub.vesoft
endif
```


Bugs Fixed Since Version 5.3

Introduction

This release fixes a number of bugs. The number in brackets at the end of each section indicates the version in which the problem has been fixed.

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]

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,

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

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]

Clearing Display Memory

When disabled, Set Visual ClearDisplay should stop the terminal or emulator from clearing its display memory. When Qedit was run from certain versions of Reflection for Macintosh, display memory was always cleared.

[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]

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]

Comment Lines in Command Files

Comment lines starting with a pound sign "#" in command files are bypassed and not executed. This means these lines can contain I/O redirection characters "<" and ">" without the risk of generating an error.

[5.3.11]