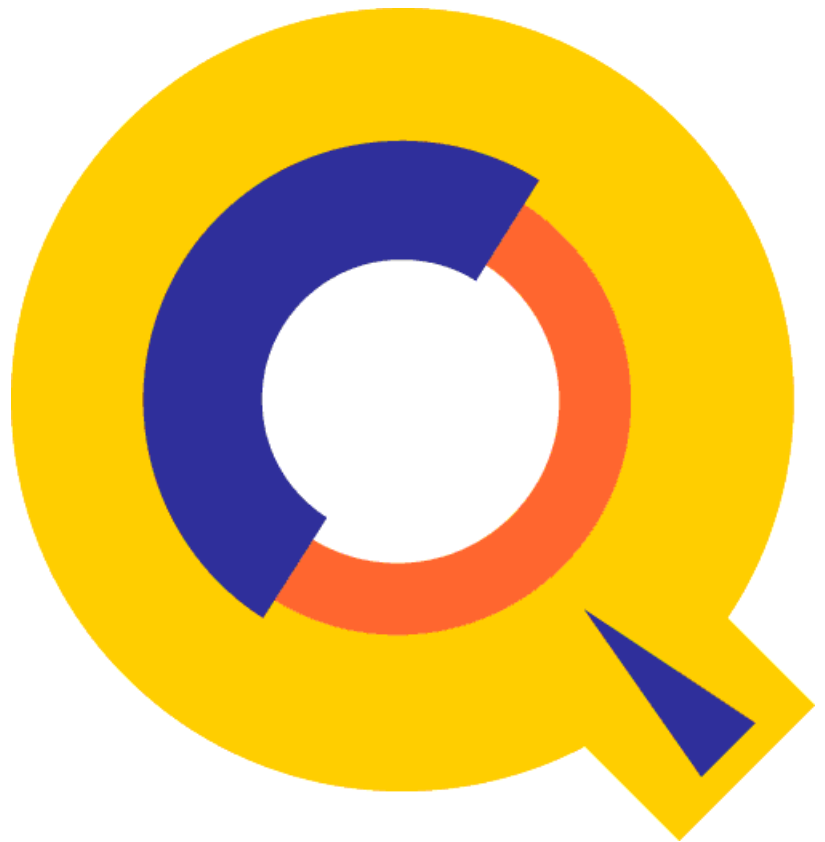

Qedit 5.5 for HP e3000

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 5.5	1
Introduction	1
Qedit for Windows	1
Highlights In Version 5.4.13.....	1
Highlights In Version 5.4.12.....	2
Highlights In Version 5.4.11.....	2
Highlights In Version 5.4.10.....	3
Compatibility	3
Unresolved Externals on MPE/iX 4.0	3
CPU Serial Number (HPSUSAN)	4
Compatibility Mode vs. Native Mode	4
Known Problems	4
Underscore in INFO= String	4
Nested Command Files.....	4
Documentation.....	5
User Manual	5
Change Notice	5
Printed Documentation.....	5
Online Documentation in HTML Help Format	5
CD or Web Download	5
Installation	7
Upgrade Instructions.....	7
Installation Assistance	7
Important Note About Passwords	7
STREAMX Users.....	8
Combined Qedit/Suprtool Tapes	8
Step 1: Install Qedit	8
Restore All Robelle Files.....	8
Create/Update Robelle Account	9
File Names.....	9
Install Program Files.....	9
Step 2: Install QLIB and Bonus Programs.....	10
Building the Spell Dictionary	10
Step 3: Install NM Compiler Interface (only XL).....	11
Installing the Interface	11
Compiling Instructions	11
Step 4: Install CM Compiler Interface (optional)	11
Choosing a CM Installation Method	12
Integrating CM Compiler Changes.....	12
Isolating CM Compiler Changes	12
Step 5: Purge Installation Files (optional)	13
Enhancements Since Version 5.4	15

Introduction.....	15
Continuous Printing and Page Ejects.....	16
Scrollup Character and Function.....	16
Merge Lines Horizontally.....	17
Preserving Trailing Spaces.....	18
Variable Substitution.....	18
Invalid Line Numbers.....	19
Cobol File Extension.....	19
Form Command.....	19
Colcopy Command [COL].....	20
Colmove Command [COLM].....	22
Excluding Lines From Visual Mode Display.....	25
Set Visual XX.....	25
Qedit for Windows	27
Introduction.....	27
Server Process.....	27
Logon Sequence.....	27
Firewall Protocol and the Server.....	28
QEDSERVMODE JCW.....	28
Bugs Fixed Since Version 5.4	29
Introduction.....	29

Introducing Qedit 5.5

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 5.4.13

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

- When loading UDC files, Qedit expected UDC names to start in column 1. This caused some UDCs to be missed and caused other commands to be used as UDCs. The MPE CI does not have the same behavior. It simply ignores leading spaces. Qedit has been changed to do as the CI.
- 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).

Highlights In Version 5.4.11

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

- Qedit can substitute environment variable names such as !HPDATEF, !HPACCOUNT used in commands. This feature needs to be enabled using **Set Varsub On**.
- By default, Qedit removes 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, 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.

- 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

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

- 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 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.

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

Underscore in INFO= String

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.

Nested Command Files

Starting with Qedit 5.4.11, complex command files with multiple levels of nesting can cause an abort. In some of the cases we have seen, the problem only occurred when the **Defer** option was used. The workaround is simply to disable the feature

with **Set Open Defer Off**. We are investigating the problem and will fix it as soon as possible.

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 5.5.

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
qcomppl	NM compiler interface
qedify, qcompusl	CM compiler interface
qloadxl	NM Qedit options
ederrnm, ederrcm	:Editor 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 Since Version 5.4

Introduction

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

- 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]
- Qedit can substitute environment variable names such as !HPDATEF, !HPACCOUNT used in commands. This feature needs to be enabled using **Set Varsub On**. [5.4.11]
- By default, Qedit removes 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, 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]

- 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]

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.

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 maximum 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 EEditonopen 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 EEditonopen 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 Trailingspaces 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 removes 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 trailingspaces 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 AFILE.GROUP.ACCT
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.

Variable Substitution

When the new `Varsub` 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. 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 exclamation point "!". For example, !HPACCOUNT is replaced with the current value of the HPACCOUNT system variable. Since the exclamation point is normally a valid Qedit string delimiter (**Verify Stringdelimiter**), Set Varsub On automatically removes it from the string delimiter list.

If you wish to prevent variable substitution and have Qedit interpret the exclamation point at face value, put 2 exclamation points as in !!HPACCOUNT.

The option is disabled by default.

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]

Cobol File Extension

Qedit looks at different file attributes to try and determine the type of file it's dealing with. Cobol source files on MPE can be identified by filecode (EDTCT) or the file extension (.cob, .cbl, .CBL) in the case of POSIX files. 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.

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

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

Self-describing files on MPE have a special filecode, SD. The data description information is stored in the file userlabels. For example, a self-describing file would look like this:

```
ACCOUNT=  GL          GROUP=  DATA
FILENAME  CODE  -----LOGICAL RECORD-----  ---SPACE---
          SIZE  TYP          EOF          LIMIT R/B  SECTORS #X MX
X1DSBEDR SD      302B  FA          3500          3  13    320  1  1
```

The Form output looks like this:

```
Self-describing information for X1DSBEDR.DATA.GL
File: X1DSBEDR.DATA.GL (SD Version B.00.00)
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
Limit: 10000 EOF: 3500 Entry Length: 44 Blocking: 64
```

LP listing

Overrides default output to \$stdlist. \$lp, \$lpa and \$lpb send output to a file with the same name as the option.

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 immediately 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 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 nonservice mode, which are not exactly the same, we provide a JCW called QEDSERVMODE. QEDSERVMODE is set to zero (0) if you are in nonservice mode, and nonzero if you are in service mode. If you are using the Qedit server, you should modify your Qeditmgr file so that few, if any, commands are executed in service mode. For example,

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


Bugs Fixed Since Version 5.4

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.

- 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]
- 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]
- 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]
- When loading UDC files, Qedit expected UDC names to start in column 1. This caused some UDCs to be missed and caused other commands to be used as UDCs. The MPE CI does not have the same behavior. It simply ignores leading spaces. Qedit has been changed to do as the CI. [5.4.13]
- **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]