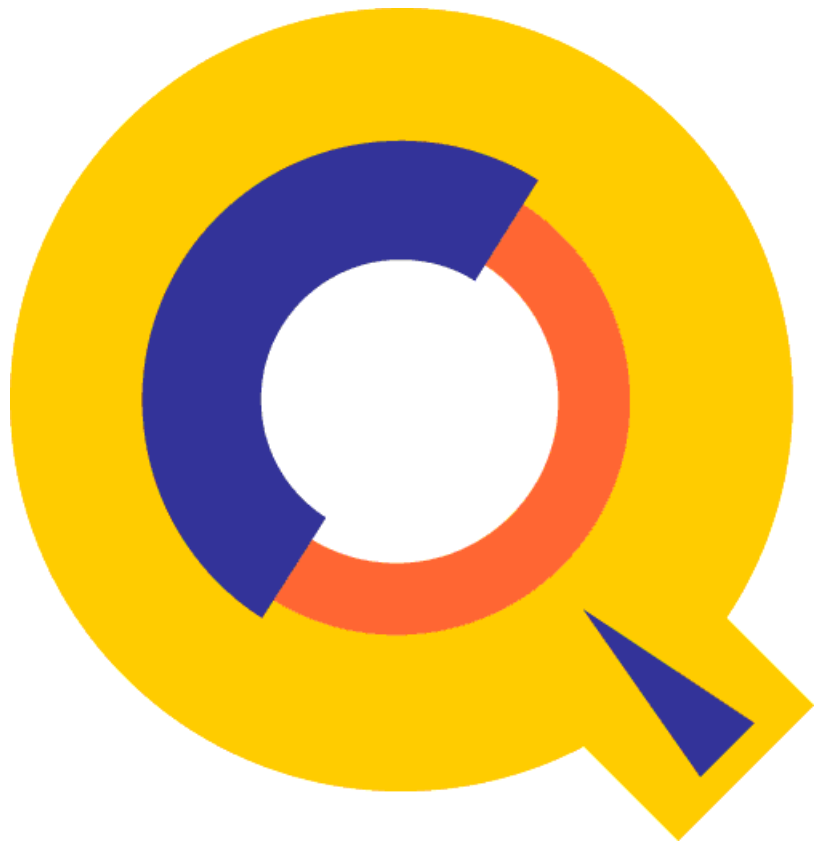


---

Qedit 6.1 for HP e3000

# Change Notice

by Robelle Solutions Technology Inc.



Program and manual copyright © 1977-2016 Robelle Solutions Technology Inc.

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

Updated January 2016

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](mailto:sales@robelle.com)

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

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

# Contents

<b>Introducing Qedit 6.1</b>	<b>1</b>
Introduction.....	1
Qedit for Windows .....	1
<b>Highlights</b>	<b>2</b>
Highlights In Version 6.1 .....	2
Highlights In Version 6.0 .....	2
Highlights In Version 5.9 .....	2
Highlights In Version 5.8 .....	2
Highlights In Version 5.7 .....	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 .....	5
Documentation.....	5
User Manual .....	5
Change Notice .....	5
Printed Documentation.....	5
<b>Installation</b>	<b>7</b>
Upgrade Instructions.....	7
Installation Assistance .....	7
Important Note About Passwords .....	7
STREAMX Users.....	8
Combined Qedit/Suprtool Tapes .....	8
Step 1: Install Qedit .....	9
Restore All Robelle Files .....	9
Create/Update Robelle Account .....	9
File Names.....	10
Install Program Files .....	10
Step 2: Install QLIB and Bonus Programs .....	11
Building the Spell Dictionary.....	11
Step 3: Install NM Compiler Interface (only XL) .....	12
Installing the Interface.....	12
Compiling Instructions.....	12
Step 4: Install CM Compiler Interface (optional).....	13
Choosing a CM Installation Method .....	13
Integrating CM Compiler Changes .....	14
Isolating CM Compiler Changes.....	14
Step 5: Purge Installation Files (optional) .....	14

<b>Enhancements Since Version 5.6</b>	<b>15</b>
Introduction.....	15
<b>Qedit for Windows</b>	<b>17</b>
Introduction.....	17
Server Process.....	17
Logon Sequence .....	17
Firewall Protocol and the Server .....	18
QEDSERVMODE JCW .....	18
<b>Bugs Fixed Since Version 5.6</b>	<b>21</b>
Introduction.....	21

# Introducing Qedit 6.1

---

## 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

---

## Highlights In Version 6.1

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

- The Verify command incorrectly parsed certain entries near the end of the verb table such as ZZ, String etc.
- The Calendar intrinsic is being phased out of Qedit and all Robelle products, which will help all products run past 2028.

---

## Highlights In Version 6.0

- UDC's would stop working properly if a Calc command longer than 138 characters was entered.

---

## Highlights In Version 5.9

- UDC's would stop working properly if a Calc command longer than 138 characters was entered.

---

## Highlights In Version 5.8

- Text command. The text command would fail with Integer Divide by Zero error when texting in some variable length or bytestream files.
- Regular Expressions. Qedit would improperly report the error "String longer than maximum allowed" if the expression specified is longer than the window of the rangelist.

---

## Highlights In Version 5.7

- MPE commands with long Info= string do not cause stack corruption anymore.
- A rangelist can now have an AND keyword with up to 10 search strings. All strings must be found on a single line for the line to be selected. Each string can have its own search window.

The following enhancements have been implemented in the Qedit for Windows server.

- The server correctly returns error 90 when the modified record length is greater than the maximum allowed in the current workfile.

---

## 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 files. You can download these from our Website.

### 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>.



# 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 5.7.

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 5.6

---

## Introduction

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

- A rangelist can now have an **AND** keyword with up to 10 search strings. All strings must be found on a single line for the line to be selected. Each string can have its own search window. [5.6.10]

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.



# 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 5.6

---

## 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.

- MPE commands with long Info= string do not cause stack corruption anymore. [5.6.10]
- UDC commands would stop working after a calc command greater than 138 characters. (5.9)
- Regular Expressions. Qedit would improperly report the error “String longer than maximum allowed” if the expression specified is longer than the window of the rangelist. (5.7.10)
- Text command. The text command would fail with Integer Divide by Zero error when texting in some variable length or byte stream files. (5.7.11)