

Installing the GPL Version of OpenQM on Ubuntu

Version: 2.00

Updated: 9 January, 2013

General

This is a guide for installing the GPL version of OpenQM onto the Ubuntu distribution of Linux. It is aimed at novice users of linux. Accordingly, I've tried to detail most things I've done to get things working. Hopefully, this will help others get things going.

Note that these instructions deal mostly with the installation of OpenQM on a Ubuntu-based distribution of Linux. As Ubuntu is based on Debian, the instructions here should be useful for anyone trying to install on Debian. A link to install notes for Fedora or Red Hat is given at the end of this document.

Likewise, because a number of other distributions are based on Ubuntu, these notes should be useful for those other distributions. These include Mint and Bodhi.

Environment

These instructions are based on a fresh install of Lubuntu 12.10 desktop edition (32 bit) running within Oracle VirtualBox on a Windows 7 based notebook. The Virtual Machine is named **VQM**, and is allocated the default 512 MB of ram and 8.00 GB of disk space recommended by VirtualBox. The disk is partitioned by the Lubuntu installer using the default settings.

Linux Editors

There are many text editors available for Linux, and each distribution comes with its own default text editor. The full Ubuntu desktop uses **gedit**; Xubuntu uses **mousepad**; Lubuntu uses **leafpad**; other distributions may use **nano**, **vi**, **vim**, **pico**, or something else. This document shows the use of **leafpad**. Use your own editor wherever this document shows **leafpad**.

Setup

We need to make sure that a few development libraries are installed on the system. These can be installed either via the 'Synaptic Package Manager' or via the command-line. The command-line options are shown here, but you should simply be able to search for the packages within Synaptic if you wish to do things that way.

Open a terminal session and type in:

```
sudo apt-get install build-essential linux-headers-`uname -r` gcc ssh
```

Some of these packages may already be installed.

If you are running your system in a virtual machine, rebuild your guest additions once you have installed the above packages.

The OpenQM help files are in the form of web pages, so we need a web server to serve these pages to client systems. To do that, we'll install Apache:

```
sudo apt-get install apache2
```



If you want any folders on the Linux system to be available to Window client systems, then you may need to install (and configure) samba also.

Networking and User/Group Setup

From an external computer, make sure that you can log onto the Linux system using putty or some other terminal emulator such as Accuterm. At this point, we need to use the ssh protocol (port 22). If you can't log in, then you won't be able to communicate with OpenQM unless you physically do all your work on the server.

We should set up a user group for all users of the OpenQM system. We will call this group 'qmusers'. From a terminal session, type:

```
sudo addgroup qmusers
sudo adduser username qmusers
```

Any user who will use OpenQM should be made a member of the qmusers group.

Make sure that you add **root** to the **qmusers** group!

There is more information on users and groups here:

http://www.techotopia.com/index.php/Managing_Ubuntu_Linux_Users_and_Groups

User and Group Permissions

If only person uses the OpenQM installation, then you will probably not have any problems with user or group permissions. However, getting permissions right for multiple users can take some time. One approach is outlined below:

Check to see what the current **umask** value is set to. From a terminal session, type:

```
umask
```

Note: Try this from your ssh session too.

If the response is 0002 (or 002), then you don't need to do anything. If it is 0022 (or something else), try the following:

```
cd /etc/pam.d
sudo leafpad common-session
```

Add the following line to the end of the profile (or change the existing line):

```
session optional pam_umask.so umask=002
```

Remove any other umask settings from other files (libpam should set umask for all login types). Check the following files:

```
/etc/profile
/home/username/.profile
/home/username/.bashrc
```

If no **umask** value is set, then leave the profiles alone as they will get the **umask** value from the default setting.

This **umask** value will create new folders and files with the following permissions:

```
Folders    775    drwxrwxr-x
Files      664    -rw-rw-r--
```



The problem with **umask** is that it sets permissions on a broad basis – all group permissions set by each user will be set as 6 (files) or 7 (folders), rather than each group (or each combination of user and groups) having their own specific default file permissions. This can mean that either:

- users give excessive default permissions to all members of their groups to allow full permissions for only some groups; or
- default permissions are set at a more restrictive level (a **umask** of 0022 will give default permissions of 755 and 644 for folders and files respectively) leading to continual manual adjustment of permissions for selected groups.

Setting **umask** as described above is probably an adequate solution for a small group of users, or for a server which only holds the OpenQM database. However, a multi-purpose server serving a large number of users will probably require a finer-grained security solution. Access control lists are one means of providing finer-grained security – google **setfacl** and **getfacl** for more information.

OpenQM

You can get the OpenQM source code from:

<http://www.openqm.com/cgi/lb.cgi.exe?X=tbl5acqe8i&t0=gpl>
http://billabong-services.co.uk/anji/qm/qmsrc_2-6-6.tgz
http://www.rushflat.co.nz/files/qmsrc_266_patched.tar.gz

You can get the matching help files from:

http://billabong-services.co.uk/anji/qm/qmhelp_2-6-6.zip
http://www.rushflat.co.nz/files/qmhelp_2-6-6.tar.gz

and the documentation from:

http://www.rushflat.co.nz/files/qmdocs_2-6-6.tar.gz

Download the GPL version of OpenQM from one of the listed locations. Note that the conditions pertaining to the GPL version are listed at the OpenQM website. You should view these even if you do not download from that location.

The GPL version of OpenQM is version 2.6-6. This is well behind the commercial version which is now at 3.0-5. The commercial version has had significant improvements over the GPL version, and is recommended if you are running a commercial site, or require support.

Save the downloaded files to a folder on the Linux system. In this case, they were saved to /home/brian.

Double-click on the ‘qmsrc...’ file from within the file manager, and it will be opened by the Archive Manager. Click on the ‘Extract’ button, then make sure the ‘Keep directory structure’ option is ticked, while ‘Do not overwrite newer files’ is not ticked, then click on ‘Extract’.

Close the Archive Manager. You should see a new folder named ‘gpl.qmsys’.

Open the folder named ‘gpl.qmsys’ and find the file named ‘buildgpl’. Double-click on this file, and choose ‘Display’ from the options presented. The file will be opened in the editor.

Find the line starting with:

```
QM_GCC_OPTIONS=
```



Make copies of this line and the QMLIBS line and comment out the originals out with a '#'. Update the two lines as follows:

```
#QM_GCC_OPTIONS="-Wall -Werror -DLINUX -D_FILE_OFFSET_BITS=64 -DGPL -g"
#QMLIBS="-lm -lcrypt -ldl"
QM_GCC_OPTIONS="-Wall -DLINUX -D_FILE_OFFSET_BITS=64 -DGPL -g"
QMLIBS="-Wl,--no-as-needed -lm -lcrypt -ldl"
```

Save the file.

There are a number of patches which should be applied to the source code. These are detailed at:

<http://billabong-services.co.uk/ANJI/qmbugs>

These patches have already been applied to the version named 'qmsrc_266_patched' that is available on the rushflat website.

Once the patches have been applied, open a terminal session, and execute the build script:

```
cd /home/brian/gpl.qmsys          Change the path name as necessary
./buildgpl
```

The build process should go through 9 stages. Some warnings will be displayed in stages 1, 3, 5, and 6. These warnings are for:

- variables being set but not used
- labels defined but not used
- differences in signedness
- an argument type mismatch in socket.h.

You can ignore these errors.

If you build the system again, you will get an additional error in stage 9, where it will warn that it cannot create a directory because it already exists. Once again, ignore this.

Now we need to move this folder structure to its proper place:

```
cd /home/brian          Change the path name as necessary
sudo mv gpl.qmsys /usr/qmsys
```

Note that this has changed the name of the folder to 'qmsys' during the move. Use the file manager to check that the folder is in place. Now, go to the /usr folder in terminal and change the group ownership:

```
cd /usr
ls -l          This will display current ownership
sudo chown -R root qmsys
sudo chgrp -R qmusers qmsys
sudo chmod -R 775 qmsys
ls -l          Check group ownership and permissions
```

Now, create a configuration file:

```
cd /etc
sudo leafpad qmconfig          Change the editor name as necessary
[qm]
QMSYS=/usr/qmsys
GRPSIZE=2
NUMUSERS=8                    Change this as necessary
SORTMEM=4096                  Enlarge the default sort memory
```

Save the configuration file. Now start OpenQM:

```
cd /usr/qmsys
```



```
sudo ./bin/qm -start
      QM has been started
./bin/qm
```

You should see the following message:

```
[ OpenQM Rev 2.6-6   Copyright Ladybridge Systems, 2007 ]
```

```
This program is free software and is supplied with ABSOLUTELY NO WARRANTY.
You are welcome to modify or redistribute this software subject to certain
conditions.  For details type CONFIG GPL.
```

This confirms that OpenQM is running. Log off by typing **OFF** at the command prompt.

Now, test that you can log in from an external machine using putty or some other terminal emulator:

```
ssh vqm          Change machine name as necessary
login as: brian  Change user name as necessary
brian@vqm's password: xxxxxx
Welcome to Ubuntu 12.10 (GNU/Linux 3.5.0-21-generic i686)

cd /usr/qmsys
./bin/qm
```

Once again, this should leave you at the colon prompt. Check that some basic commands work, and create a **MASTER.LOGIN** item.

```
LISTU
LISTFL
SH pwd
PTERM DISPLAY
ED VOC MASTER.LOGIN
  I          Put the editor into insert mode
  PA        Paragraph
  DATE.FORMAT ON  OFF for North American users
PTERM CASE NOINVERT
<enter>      Exit from insert mode
  FI        File the MASTER.LOGIN item

QUIT
exit
```

Configuration

Automatic Startup

Create a script in the /etc/init.d folder as follows:

```
cd /etc/init.d
sudo cp skeleton qm          Copy the skeleton script to 'qm'
sudo leafpad qm             Edit new qm script
```

```
#!/bin/sh
### BEGIN INIT INFO
# Provides:          OpenQM
# Required-Start:
# Required-Stop:
# Default-Start:    2 3 4 5
# Default-Stop:     S 0 1 6
# Short-Description: Initscript for OpenQM
# Description:      Starts and stops OpenQM at startup/shutdown
#
### END INIT INFO

# Author: Brian Speirs
```



```

#
# Adapted from skeleton script file
#

PATH=/sbin:/usr/sbin:/bin:/usr/bin:/usr/qmsys/bin
DESC="OpenQM database"
NAME=qm
DAEMON=/usr/qmsys/bin/qmlnxd
PIDFILE=/var/run/$NAME.pid
SCRIPTNAME=/etc/init.d/$NAME

# Exit if the package is not installed
[ -x "$DAEMON" ] || exit 0

case "$1" in
  start)
    echo "Starting OpenQM database." >&2
    qm -start -quiet
    ;;
  stop)
    echo "Stopping OpenQM database." >&2
    qm -stop -quiet
    ;;
  restart|force-reload)
    echo "Restarting OpenQM database." >&2
    qm -restart -quiet
    ;;
  *)
    #echo "Usage: $SCRIPTNAME {start|stop|restart|reload|force-reload}" >&2
    echo "Usage: $SCRIPTNAME {start|stop|restart|force-reload}" >&2
    exit 3
    ;;
esac

:

```

Save the modified script.

```

sudo chmod +x qm
sudo update-rc.d qm start 75 2 3 4 5 . stop 25 0 1 6 .

```

These first of the two steps above makes the script executable, while the second creates symbolic links in the startup folders so that OpenQM will be started and shut down automatically. Make sure you include the two periods in this second command (after the 5 and the 6). The startup priority is set at 75 (relatively late), while shutdown priority is 25 (relatively early). You may change these priorities as you see fit, or simply use the default priorities (20 and 80):

```

sudo update-rc.d qm defaults

```

Test the script:

```

sudo ./qm stop
Stopping OpenQM database
QM has been shutdown
sudo ./qm start
Starting OpenQM database
QM has been started

```

If you don't get the messages appearing after your commands, then check the script for errors.

Make available to users

Now, put the qmsys/bin folder into the \$PATH variable for all users:



```
cd /etc
sudo leafpad environment
```

Add the following path to the list of paths in this item:

```
/usr/qmsys/bin
```

Create a “home” directory

This is based on the assumption that you want to keep the OpenQM data files under the home directory, but separate from other users.

Open a terminal session, and create a folder under the /home folder named ‘qm’:

```
cd /home
sudo mkdir qm
sudo chgrp -R qmusers qm   Make qmusers the group associated with the qm folder
sudo chmod -R 775 qm       Give owner and group read/write/execute permissions
sudo chmod -R g+s qm       Ensure the group retains permissions for new files
```

This sequence should leave you as the owner of the folder, but with the group set to ‘qmusers’. Further, any new file or folder created within the ‘qm’ folder should belong to the ‘qmusers’ group.

Restart Linux

Restart Linux and check that preceding changes are working correctly. After restarting:

```
ps -ef | grep qm
```

This should display two lines – one showing the qm daemon (qmlnxd), and the other showing the grep command. If the qm daemon is not running, go back to the automatic startup script and try to figure out why not.

Now check the qmsys/bin folder is in your path:

```
echo $PATH
```

You should see /usr/qmsys/bin in the list of folders reported back.

Log in with AccuTerm

This assumes that you will use AccuTerm (on a Windows PC) as the terminal emulator to log into OpenQM.

Start Accuterm, then click on the ‘Create a new session’ icon (or choose ‘File | New’ from the menu). Click on the ‘Connection Wizard’ button.

Select ‘Secure Shell’ as the connection type, then click ‘Next’.

Enter the name of your Linux machine in the ‘Host name’ box, leave the host type blank (no options available) and leave the port number as ‘default’. Click on ‘Next’.

Choose a terminal type on the next screen (VTxxx terminals work well in Linux), then click ‘Finish’. A login prompt should appear. Log in.

Change to the qmsys folder, and check that you can access OpenQM:

```
cd /usr/qmsys
qm
```

Now create a new account for the ACCUTERM host programs:



```

:CREATE.ACCOUNT ACCUTERM /home/qm/accuterm
  Create new directory for account (Y/N)? y
  Creating VOC...
  Creating $HOLD...
  Creating $SAVEDLISTS...
  Creating private catalogue directory...
  Adding to register of accounts...
:LOGTO ACCUTERM
:WHO
  2 ACCUTERM from QMSYS
:!pwd
  /home/qm/accuterm
:OFF

```

Now, it is up to you!

Services

At this stage, to use OpenQM, you need to log onto the Linux server, change to an appropriate directory and then invoke the qm executable. However, the normal procedure is simply to log onto an OpenQM service (without actually logging into Linux). Likewise, we cannot use qmclient connections to create native applications – we only have terminal emulation.

We can log directly into OpenQM using the ‘qmsrvr’ service. The catch is that we then need to use the Telnet protocol which is less secure than ssh. On the other hand, we will be using port 4242 for the telnet connection, so the connection is obscure (rather than truly secure).

First we need xinetd. We may as well install telnet now too.

```
sudo apt-get install xinetd telnetd-ssl
```

The install process will automatically start the xinetd service.

Now, place the following service definitions in the /etc/xinetd.d folder:

```

cd /etc/xinetd.d
sudo leafpad qmsrvr

# Description: Service allowing connections to the OpenQM
#               database server from telnet clients.
#
service qmsrvr
{
    protocol          = tcp
    flags              = REUSE
    socket_type        = stream
    wait               = no
    groups             = yes
    user               = root
    group              = qmusers
    umask              = 002
    server             = /usr/qmsys/bin/qm
    server_args        = -n
    log_on_failure    += USERID
    disable            = no
}

sudo leafpad qmclient

# Description: Service allowing QMClient connections to
#               the OpenQM database server.
#

```



```

service qmclient
{
    protocol      = tcp
    flags         = REUSE
    socket_type   = stream
    wait          = no
    groups        = yes
    user          = root
    group         = qmusers
    umask         = 002
    server        = /usr/qmsys/bin/qm
    server_args   = -n -q
    log_on_failure += USERID
    disable       = no
}

```

Note these service definitions define the user, group, and umask settings for the services. These affect file permissions of any files created, and it is important to get these right. Similarly, because root is the ‘user’ of these services, it is important that root is a member of qmusers (see: Networking and User/Group Setup).

Now we need to tell xinetd about these services. Add the following two lines to the services file:

```

cd /etc
sudo leafpad services

qmrvr      4242/tcp      # QMSrvr
qmclient   4243/tcp      # QMClient

```

Restart xinetd to put these changes into effect:

```

sudo stop xinetd
sudo start xinetd

```

You should now be able to log directly into the OpenQM environment from an external machine using telnet on port 4242 without changing directories and invoking the qm executable.

File Ownership and Permissions

You may well strike problems with file ownership and permissions. These take time to get right.

Error numbers 3018 (file is read-only) or 3035 (permissions error) are the most likely. Error 3018 will occur if you try to write or delete an item, whereas 3035 probably means you don’t have permission to read the file.

Troubleshooting steps:

- Check who owns the file, and what group is associated with that file.
- Check who you are (type **!whoami** from the OpenQM prompt, or **whoami** from the Linux prompt), then check which group you are in (use the ‘Users and Groups’ manager).
- Check your umask settings (type **!umask** from the OpenQM prompt, or **umask** from the Linux prompt).
- Try creating a file and checking what permissions are given to that file.
- Try logging in as a different user, and see if that user has file permission problems.



For further information on Linux file and folder sharing, look here:

<http://www.udel.edu/topics/os/unix/general/groupsharing.html>

Help and Documentation

Download the help and documentation files from one of the locations specified on page 3 of this document. For our purposes, we'll assume they are in /home/brian.

Unzip the files to create the folders and files.

From a terminal session, type:

```
cd /home/brian           Change to match actual location
mv qmdocs_2-6-6 /home/qm/docs
mv qmhelp_2-6-6 /home/qm/help
cd /var/www              Move to web root
sudo mkdir qm            Make a qm folder
cd qm
sudo mkdir help          Make a help folder
cd help
sudo ln -s /home/qm/help index.html  Create a link back to the help folder
```

Now, from an external machine, point your browser at:

```
http://vqm/qm/help      Change machine name as necessary
```

The OpenQM help system should open up in your browser.

Once everything is running OK, you can delete the original zipped versions of the help files and documentation.

What Do I Do Now?

If you are already familiar with multi-value databases, then you will be able to proceed confidently from here. On the other hand, if you haven't used a multi-value database before, then you need some help. Luckily, there is some available!

There are three resources that will get you started, and get you to the point where other material starts to make sense:

- In the documentation that you have installed, there is a tutorial on using OpenQM. Open this using your PDF reader.
- Go to the Downloads page of the OpenQM web site (www.openqm.com), find the 'Documentation' section, and download the 'Teach Yourself OpenQM' guide.
- Go to www.rushflat.co.nz and download the two 'Getting Started in OpenQM' books.

Once you are familiar with the system, look through general multi-value resources (such as PickWiki or the MVDBMS google group) to gain further knowledge.

Concluding Remarks

At this stage, OpenQM should be installed on the Linux server, and set to start and shutdown as the server changes run levels. You should be able to access OpenQM from an external machine using telnet, ssh, or a qmclient connection (not tested here); and the file permissions should be set correctly to allow OpenQM to operate (provided users are in the 'qmusers' group).

What you do now is up to you.

Other OpenQM Resources

OpenQM home page: <http://www.openqm.com>
OpenQM Google groups: [http://groups.google.co.uk/group/OpenQM?hl=en&lnk=
http://groups.google.co.uk/group/openqm-opensource](http://groups.google.co.uk/group/OpenQM?hl=en&lnk=http://groups.google.co.uk/group/openqm-opensource)
<https://groups.google.com/forum/?fromgroups=#!forum/scarletdme>
Getting started in OpenQM: <http://www.rushflat.co.nz>
Scarlet DME home page: http://www.scarletdme.org/wiki/Main_Page
AccuSoft Enterprises: <http://www.asent.com>
Anji Development environment: <http://billabong-services.co.uk/ANJI/>
Install notes for Red Hat www.geneb.org/qm/fedora_notes.txt

Other Multi-value Resources

MVDBMS Google group
<https://groups.google.com/forum/?hl=en&fromgroups#!forum/mvdbms>
Pick Wiki: <http://www.pickwiki.com/cgi-bin/wiki.pl>