qmExplorer

Written by: Brian Speirs
Rush Flat Software
7 Trinity Avenue
Lower Hutt 5011
New Zealand

Support: brian at rushflat dot co dot nz

Introduction

qmExplorer is a simple explorer application for the OpenQM database. It has been tested against QM/Linux 4.0-7 (64-bit) and QM/Windows 4.0-9 (64-bit), and on clients running Linux Mint and Windows 11.

qmExplorer is a 64-bit application and therefore requires a 64-bit client environment to run in. It **might** talk to 32-bit installations of OpenQM but this has not been tested.

It is unlikely to work against the OpenQM derivatives (ScarletDME and SD) due to the coding to handle distributed files (which were not part of the GPL version of OpenQM) – but this has not been tested.

qmExplorer is simply an exploration tool and nothing more. It does not **write** anything to the OpenQM system. It will show you files and other objects on the system, but it does not modify them.

Back to top

Installation

Linux or Windows

There are two alternate installation archives – one for Linux, and one for Windows. Make sure that you have the correct archive for your **client** operating system. For example, if you have an OpenQM installation running on a Linux server, but you connect to the database from a Windows workstation, then you need the Windows archive.

In either case, you need to install the client components on each client computer that you wish to run qmExplorer from; and you need to install the server subroutine on your OpenQM server.

Client installation

The client executable does not need any installation as such. Simply unzip the distribution archive to a suitable folder on your system. Make sure that:

- the executable has ‘execute’ permissions (Linux)
- users of the program have write permissions on the folder.

Server installation

The server program needs to be installed on your OpenQM system:

- Using a text editor, copy the contents of the **`QMX.ACTION`** item in the qmExplorer folder
- Create a new item named **`QMX.ACTION`** in your programs file (e.g. BP) on your OpenQM system and paste the contents of the item you copied in the step above
- Save the file and exit
- Compile the program: **`BASIC BP QMX.ACTION`**
- The program will be automatically catalogued globally.

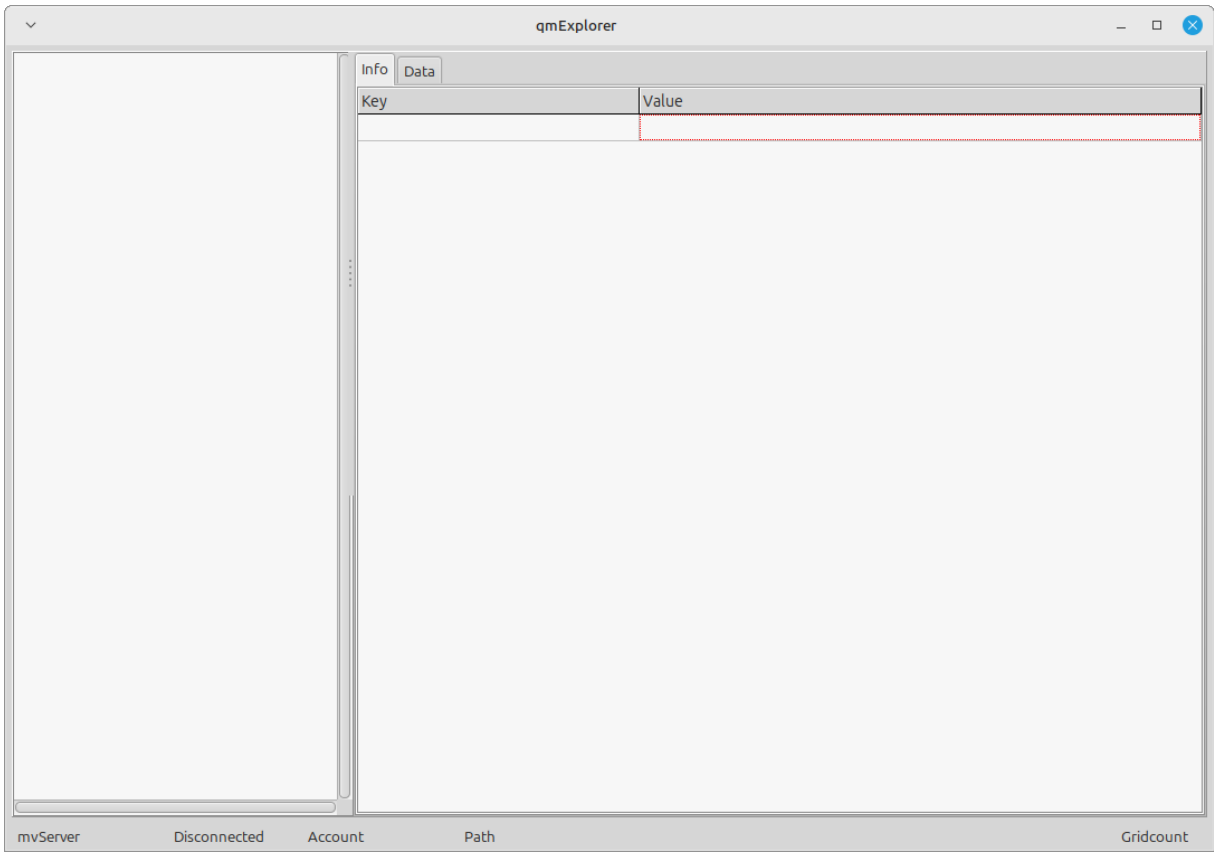
The program should now be “installed”. However, you may wish to create a shortcut from the executable to our desktop – or create an entry in your menu system to start the client application.

Back to top

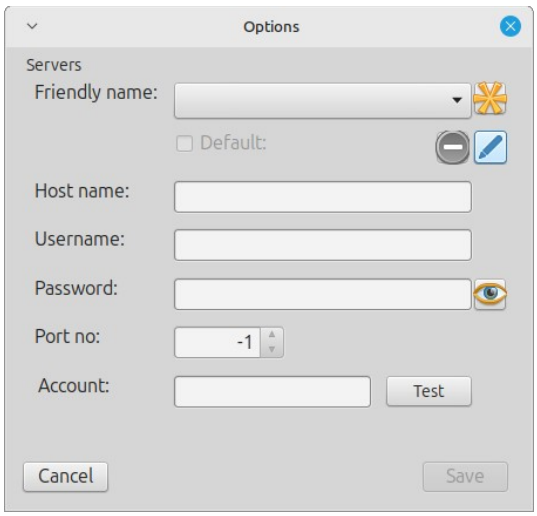
Configuring servers

qmExplorer can explore multiple servers. Connection information for these servers is encrypted and stored on the client machine.

Start by starting the executable. This should display a screen like this:



At the left end of the status bar, you will see the text ‘mvServer’. Right-click on this text and choose **Options** from the shortcut menu. That will bring up this screen:



To create a new server definition:

1. Click on the orange asterisk 

2. Fill in the fields on the form (they are all mandatory)
   a) You can leave the **Port no** field set to -1 to use the default QMClient port
3. If this will be the default server to use when opening the application, tick the **Default** checkbox
4. Once all fields have been filled in, click on the **Test** button
   a) You will be informed whether the connection was successful or not
   b) If the connection was successful, then the **Save** button will be activated
5. Click on Save to save the details and exit the form.

To edit an existing connection:

1. Select the connection from the **Friendly name** drop-down list
2. Click on the **Edit** icon
3. Update the fields as necessary
4. Click on the **Test** button to test the connection
5. Click ont the **Save** button once you get a successful connection test.

To delete an existing connection:

1. Select the connection from the Friendly name drop-down list
2. Click on the **Delete** icon
3. The connection will be deleted without any further prompts.

Back to top

Using qmExplorer

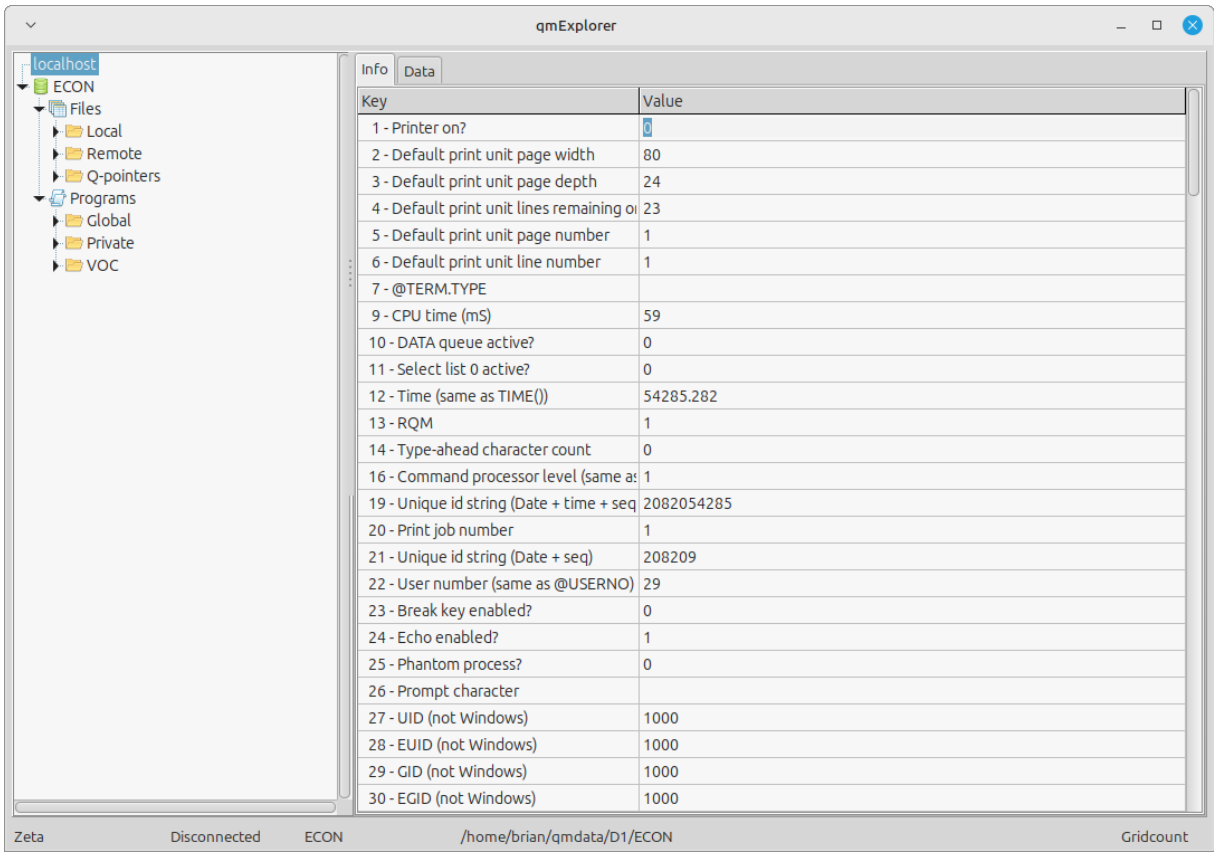
Getting help

You can press the **F1** key at any time to display this help.

Back to top

System information

Once you have configured at least one server, then the next time you start qmExplorer you should get a screen like this:



Look first at the status bar. This has five fields:

1. The **Friendly name** of the current server
2. The status of the connection to the current server
3. The account that is being explored on the current server
4. The path of the account that is being explored
5. The count of items in the data grid

In this case, the current server has a Friendly name of **Zeta**, and the connection is currently **disconnected**. We are exploring account **ECON** and it has a path of **/home/brian/qmdata/D1/ECON**. The **Gridcount** field has not been updated because we are not (yet) exploring a file.

Connections to servers are only established “as required”. The connection will time out after 30 seconds if it is not being used.

On the left of the screen is a tree view of the current system / account. The root of the tree is the **Host name** that was defined for this server.

On the right of the screen are two tabs:

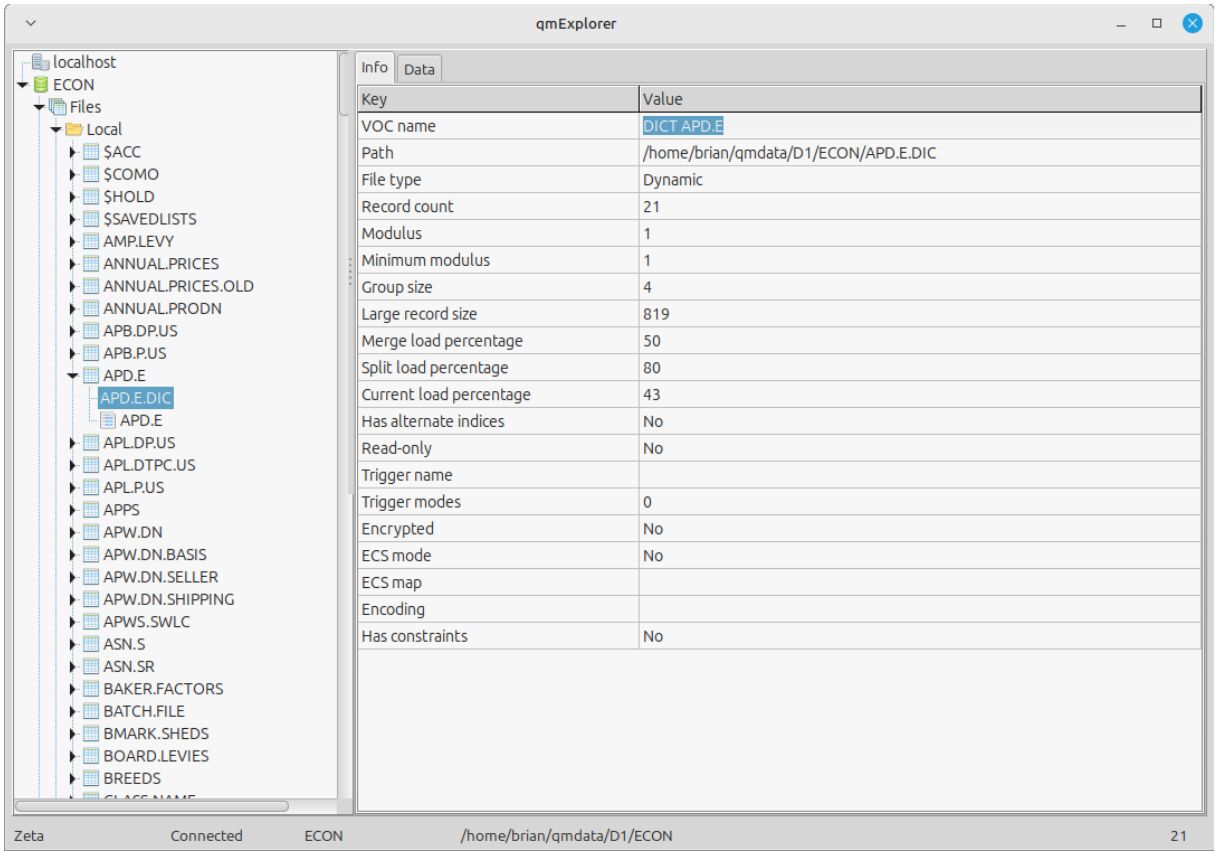
- **Info** shows information for the currently selected node of the tree
- **Data** shows (some of) the data in the currently selected file
  - If the currently selected node is not a file, then this tab will be blank

When the root node is selected, the **Info** tab shows the output from all the valid **SYSTEM()** keys. Where the return value has multiple attributes, these are spread over successive lines in the output grid. Where the return value has value marks, these are replaced by the string **<vm>** (and sub-value marks would be represented by **<sm>** if present).

Back to top

File information

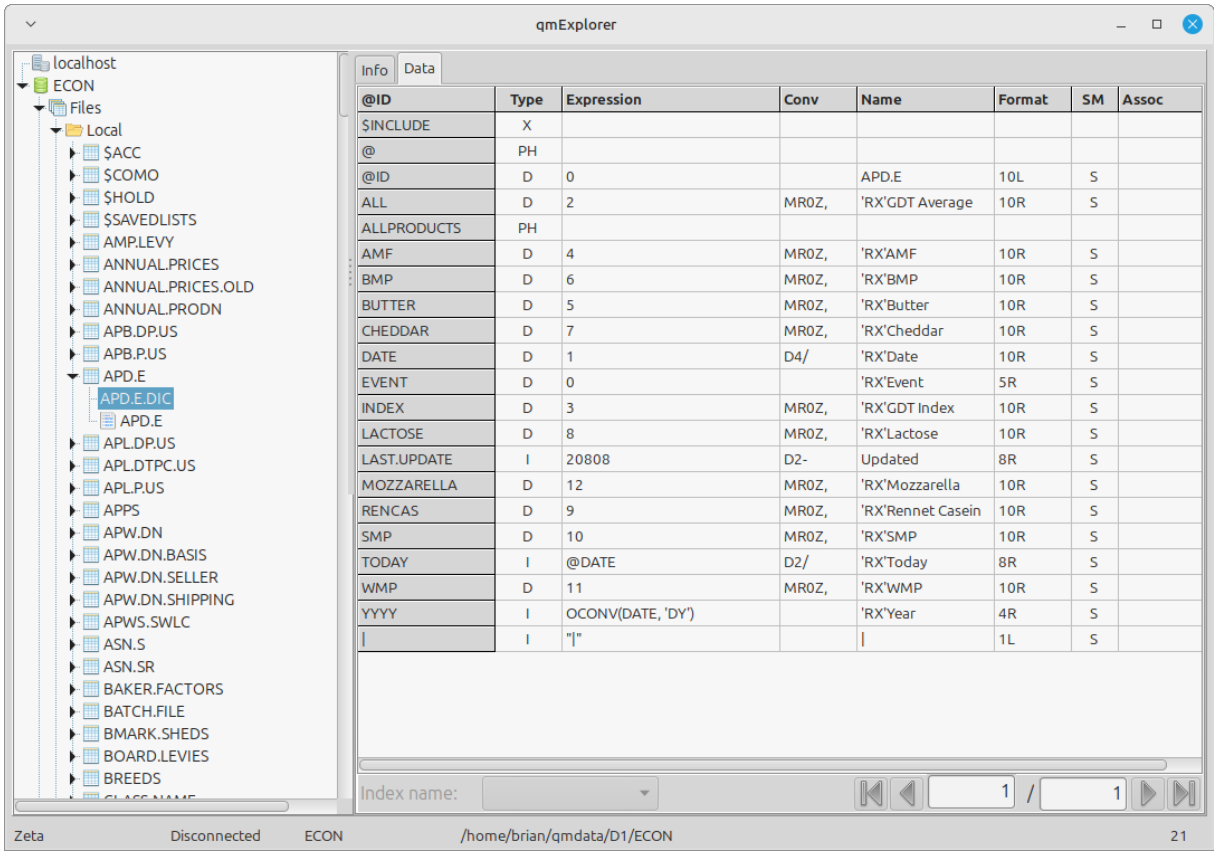
If we open the tree view to show the local files, and then select one of the files, we get a view like this:



The information displayed in the **Info** tab corresponds to selected keys from the **FILEINFO()** function. If the selected file has alternate key indices (or constraints), then the indexed (constrained) fields are also displayed.

The **Data** tab shows (selected) data from the selected file. Note the following rules that have been applied here:

- A maximum of 10,000 records will be displayed in the data grid
- If a dictionary has been selected:
  - The items will be sorted into ascending alphabetic order
  - The items will be shown in a D or I-type structure (i.e. A and S-type items are converted to this structure for display here)
- If a data file has been selected:
  - The display will use the '@' item in the dictionary to define the display fields
  - If there is no '@' item, then the dictionary will be read and the dictionaries for the first 10 dictionary items (sorted by attribute mark count) will be selected for display
  - If there is no dictionary, or there are no dictionary items in the dictionary, then a default dictionary will be used for up to 10 display fields (plus @ID)
  - If the display dictionary items are I-types and they have not been compiled, then the field will not return a value
  - Data records are not sorted
  - However, if there are indexed fields in the file
    - the initial display will use the first defined index
    - users can choose to display data using one of the other indexed fields (or as non-indexed data) by selecting the index in the drop-down list at the bottom of the screen.



If there are more than 10,000 items in the file, then the paging buttons at the bottom of the

data grid will become active. Users can use the buttons to page through the data, or enter a page number in the first page number field to go directly to that page.

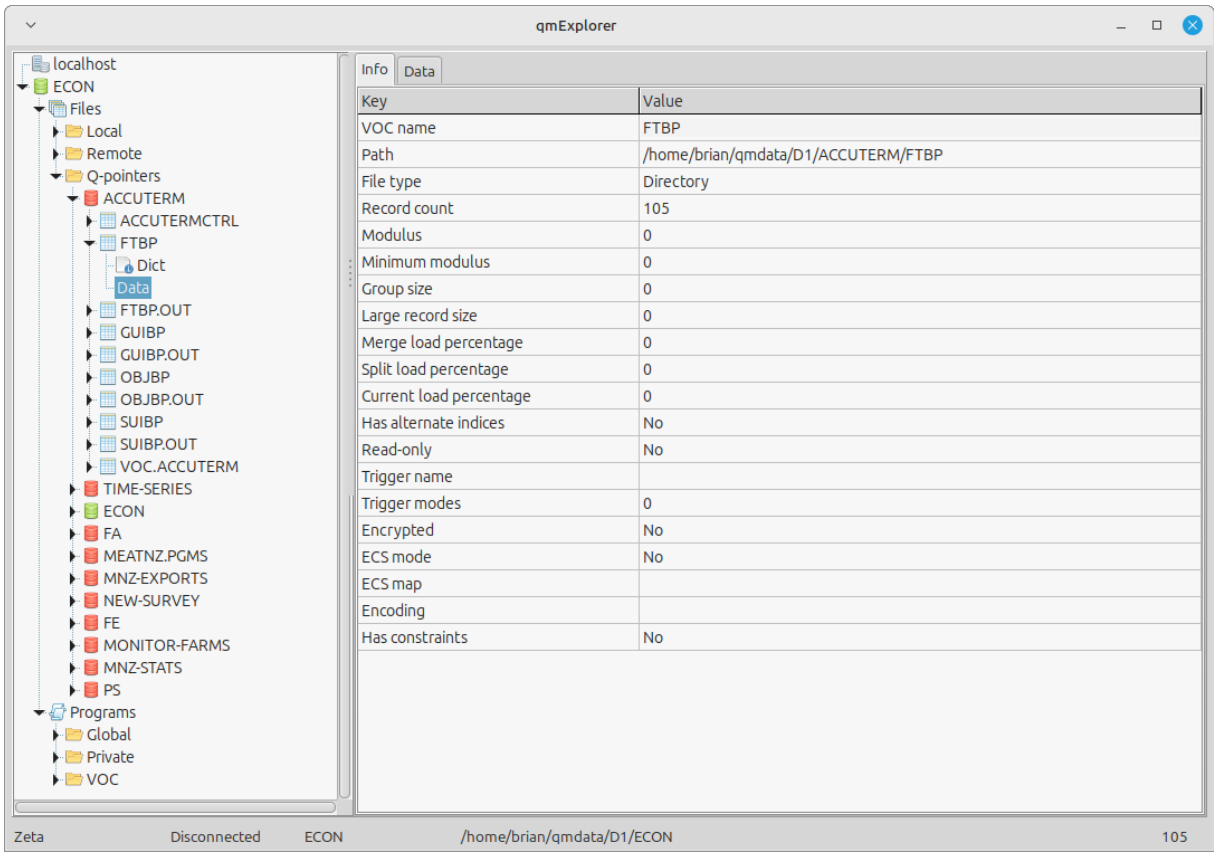
Remote files

The tree display for remote files is essentially identical to that for local files.

Back to top

Q-pointers

The display of Q-pointers in the VOC of the current account is broken into “source accounts”. Opening each source account will show you the Q-pointers in the current account that point to that account:

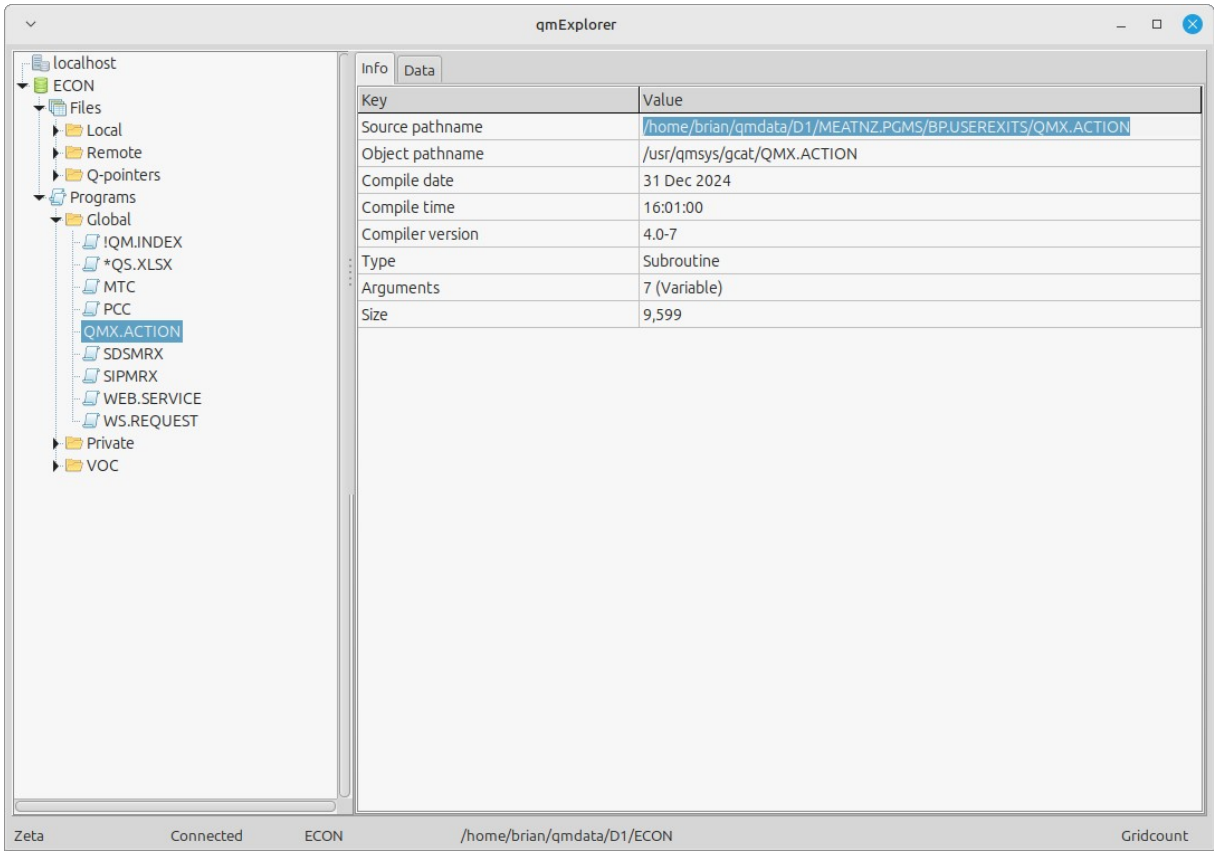


Note that the account icons are colour-coded. The green icon represents the current account while the red icons are other accounts.

Back to top

Program information

Programs are sorted into their catalogues. Selecting a program within its catalogue shows some basic information about the program:



There is no information on the Data tab for programs.

Back to top

Changing servers

To change the server that you wish to explore, right-click on the **Friendly name** displayed at the left end of the status bar and select the desired server from the list of configured servers.

If the server you wish to explore is not shown in the list of configured servers, then choose **Options** from the shortcut menu, and follow the instructions for adding new servers in the Configuring servers section.

Back to top

Changing accounts

To change the displayed account, right-click on the **Account** field in the status bar and choose the desired account from the list of accounts on this server.

Back to top

Technical notes

qmExplorer has been written in FreePascal using the Lazarus IDE.

It uses the QMClient library to communicate with the OpenQM server. In particular, it uses the **`QMCallx`** function to call the **`QMX.ACTION`** subroutine. Older versions of OpenQM may not support this function call.

Users can change the **`QMX.ACTION`** subroutine to:

- change the number of items returned
- order of returned data items
- fields returned.

Care should be taken if modifying this subroutine. Rush Flat Software takes no responsibility for damages caused by user modifications.

As written, the **`QMX.ACTION`** subroutine requires the **`QMCLIENT`** configuration parameter to be set to a value of 0 or 1. If your OpenQM system runs with a **`QMCLIENT`** configuration parameter of 2, then you should add the **`$QMCALL`** compiler directive to the **`MVX.ACTION`** subroutine and re-compile the subroutine. Note: This has not been tested.

Use with OpenQM derivatives

Subroutine **QMX.ACTION** is written in a way that ***might*** let it work with ScarletDME – but it will almost certainly ***NOT*** work with SD (String Database). These OpenQM derivatives are based on the OpenQM 2.6-6 code base which does not include:

- Encryption
- Distributed files
- ECS mode
- File constraints
- Try / Catch structures in QMBasic code

Access to these **FILEINFO** keys within the code is wrapped within **$IFDEF / $ENDIF** compiler directives. This ***should*** let the code compile and run within ScarletDME – but this has not been tested.

To activate this mode, uncomment the line in the code that defines the symbol **ISGPL** . This line should read:

```
$DEFINE ISGPL 1
```

Recompile the code:

```
BASIC BP QMX.ACTION
```

Troubleshooting

If **qmExplorer** does not connect to a remote system, check the firewall settings on both the client and server systems. This needs to allow QMClient connections. By default, the

QMClient system communicates on port 4243 – but your system may use some other port. Check with your system administrator.

qmExplorer does not need any other components installed.

Notices