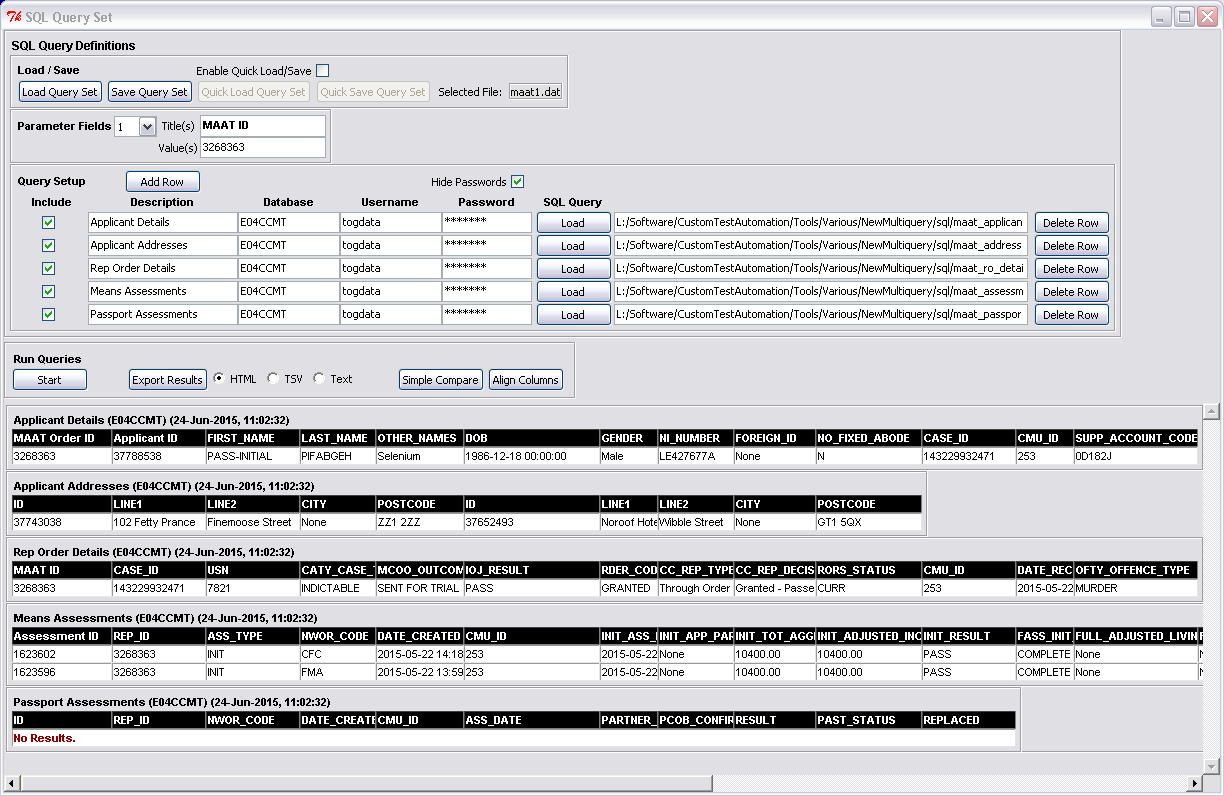
NewMultiQuery Guide

|  |  |  |  |
| --- | --- | --- | --- |
| **Author** | **Version** | **Date** | **Description** |
| Alan Maydwell | 0.1 | 24 June 2015 | Initial version |
| Alan Maydwell | 0.2 | 13 August 2015 | Added info about newly added “Reload SQL” button. |

# Introduction

This document is a brief guide to the Python Application NewMultiQuery.



This application enables the quick running of individual or multiple SQL queries.

It has the following capabilities:

* The creation, saving and loading of sets of SQL queries.
* Defining substitution parameters for the “where” clauses of the queries.
* Running set of queries.
* Displaying results on-screen.
* Exporting results as HTML tables or tab-separated variables.
* Performing simple comparisons between the query results.

# Prerequisites

Currently this application is only intended to be run on VDI PCs. As it is written in Python and makes use of the Python pyodbc module, the following needs to be installed before it can be run:

|  |  |  |
| --- | --- | --- |
| **Item** | **Source** | **Notes** |
| Python v2.7 | L:\Software\Python\python-2.7.3.msi | Earlier versions might work too. Version 3 or higher won’t work. |
| podbc | L:\Software\Python\PythonExtensions\ForPython2.7\pyodbc-3.0.6.win32-py2.7.exe | Python must be installed first |
| Microsoft ODBC driver for Oracle | ??? | Should already be installed on VDI PC. |

It might also be necessary to create an environment variable called “TNS\_ADMIN” holding the path to your TNSNAMES.ORA file.

# Running

As long as the items in the above section have been installed on the VDI, the application can be run by going to

**L:\Software\CustomTestAutomation\Tools\Various\NewMultiquery\**

and clicking **NewMultiQuery0.5.pyw** (or similar filename with different version number).

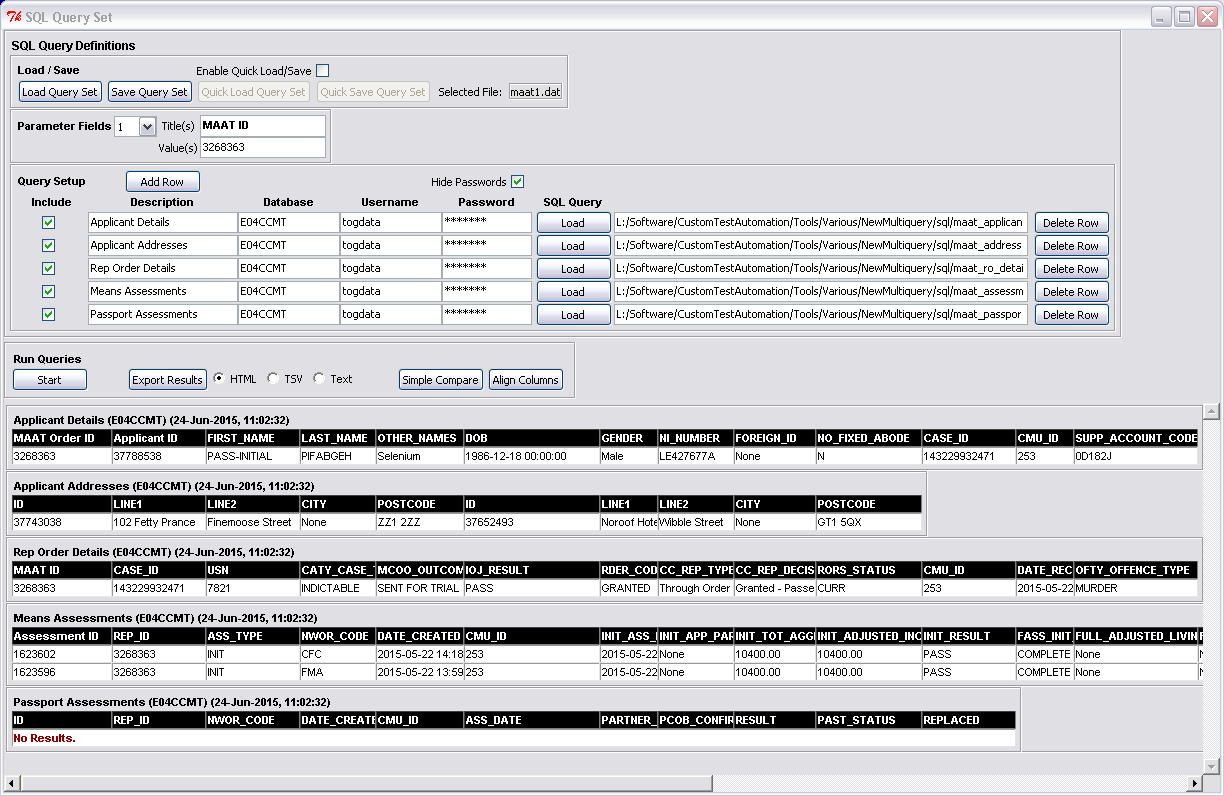
Some example query set files and this present document can be found in the same folder.

If you plan on making extensive use of this application it’s worth taking a copy and placing it in a folder of your own choice.

**Verbose Mode**

Changing the file extension from **pyw** to **py** before launching it will cause the application to open a console window in which addition details are displayed.

# Sections of the Screen



**1** Load/Save sets of queries

**2** Parameter definitions

**3** Query Definitions

**4** Run queries, export results, compare results

**5** Results

## 1 Load / Save

Sets of queries, including any associated parameter fields, can be saved and loaded using these buttons. To reduce the chances of accidentally pressing the quick load and quick save buttons, they are disabled unless the “Enable Quick Load/Save” tickbox is ticked.

* The filename in current use is displayed in the “Selected File” box. If none has been chosen, this defaults to “default.dat”.
* The saved files only store the paths to SQL used by the queries rather than the SQL itself. If one of the included SQL files is moved/deleted/renamed then the query set will need to be updated.
* The files created here are created using the Python “pickle” module and cannot be easily updated in a text editor.

## 2 Parameter Definitions

Optionally up to 6 parameter fields can be defined here. The values recorded here can be substituted into the SQL, as long as the SQL has been setup in a particular way.

The title given to each parameter field is merely an on-screen label that plays no part in how the parameter behaves. The association between the parameter and the SQL script is established by placing question marks in the script in the locations where you wish the parameter to be used, with correspondence established simply by position – i.e. first parameter with first “?”, second parameter with second “?”

For example, if the following script is used and two parameters are defined.

**SELECT \***

**FROM addresses**

**WHERE**

**Id = ?**

**and**

**date\_created > ?**

Then the first parameter on-screen will be associated with **Id** in the SQL script and the second parameter with **date\_created**.

The application will automatically only use the parameters with SQL scripts that contain question marks. Surplus parameters are ignored. In other words, the application won’t apply more parameters than there are questions marks. It is therefore fine for a single set of queries to include SQL scripts that use all, some or none of the parameters.

Warning – all question marks in the SQL script are evaluated. Take care that no extra ones have been included anywhere, including in comments.

## 3 Query Details

One or more SQL queries can be defined here.

**Add Row button** – adds an extra row. Currently no maximum is defined but the screen will become overcrowded if too many are added.

**Hide Passwords** **tickbox** – when active all the passwords are masked with asterisks.

**Include** **tickboxes**- determines whether the SQL query will be executed when the Start button is pressed. Does not affect Save Query Set which always includes all queries.

**Reload SQL button –** re-loads all the SQL files already specified in the SQL file path fields. The chosen SQL files are automatically loaded into RAM when the “Load Query Set” button or individual “Load” buttons are used, but they are not re-loaded when the “Start” button is pressed. This button is used to force a reload. This is helpful when (a) an SQL file has been amended after having been chosen, (b) an SQL file path has been manually updated within the application.

**Description Field** – Optional name given to query. Does not affect execution. Is automatically included in exported files.

**Database, Username, Password –** key connection parameters for each query.

**Load button** – Opens file selector to choose SQL file.

**Filepath field** – Displays path to selected SQL file. **Warning -** directly modifying the contents of this field makes no difference. Currently a file can only be selected by pressing the Load button.

**Delete Row button** – delete the row, but only if there’s more than one present.

**Updated SQL**

The SQL scripts used by the queries are stored in RAM and only reloaded under the following circumstances:

* “Load Query Set” button pressed.
* “Load” button used to select the SQL script.
* “Reload SQL” button pressed.

If an included SQL file is updated after it has already been loaded, the changes will not be picked up by the application unless one of the above three actions is taken.

## 4 Run Queries

Run the query set. Export the results. Perform simple comparison.

**Start Button** – Runs all queries with active “Include” tick boxes. If there is a connection problem or an error in the SQL script, an error message should be displayed in the results section of the screen and execution should skip to the next query in the set.

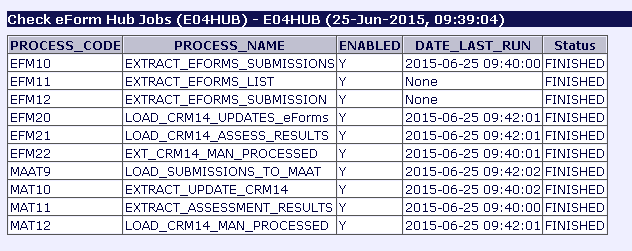
**“No Results” Error**

Some queries fail to return results and display a message saying “**No Results. The driver did not supply an error!**” even when the SQL is correct and runs without problem in Toad.

This is commonly because the Microsoft ODBC driver is unable to handle some of the data types used in our Oracle databases. For example it doesn’t like START\_TIME\_STAMP column in the HUB’s HUB\_PROCESS\_LOGS table. This problem can usually be worked around by placing a to\_char function around the offending item in the SQL. For example replace START\_TIME\_STAMP with to\_char(START\_TIME\_STAMP).

**Export Results** – Saves all currently displayed results to a file in the chosen format (HTML, TSV, Text). HTML format is the nicest to look at and includes any colour coding applied by the comparison button. TSV format can be loaded into Excel. “Text” saves all results in single column in a text file, which sometimes can be usefully used in conjunction with a file comparison tool.

Example HTML format result

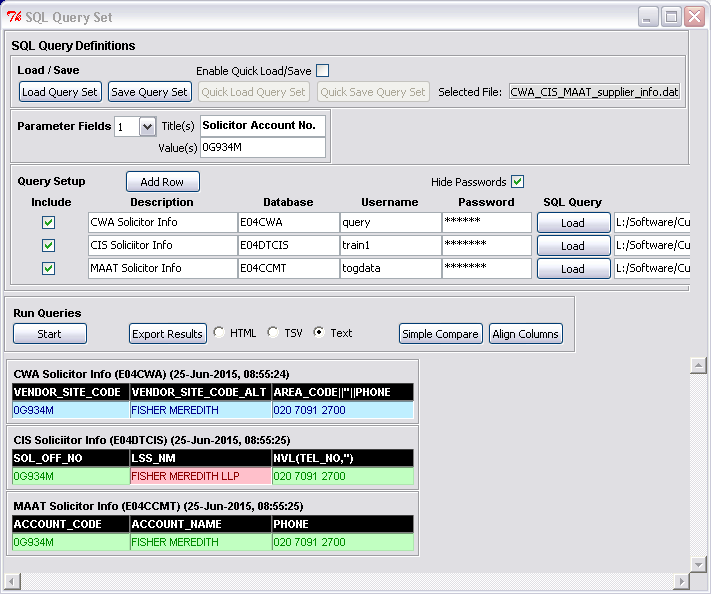


**Simple Compare Button** – Performs simple row-by-row, column-by-column comparison between all the rows in the topmost query with any corresponding rows/columns in all the other queries.

If the topmost query has rows/columns that the query it’s being compared with lacks, or the converse is true, then no check will be made of these particular cells.

Matching values are coloured in green, non-matching in red. The reference values from the top query are coloured blue.

Below is an example showing comparison of some solicitor details between CWA, CIS and CCMT.



**Align Columns –** by default the width of each column for each query is automatically adjusted to the suit the maximum width of its contents, including the column heading. This means the columns for the different queries are unlikely to line up. Pressing this button overrides the default setting and causes the column widths of each query to match those from the topmost query. This can make comparison results easier to read.