

DBS2QVLS – DataBase Search to QlikView Load Script Utility

Contents

Introduction	1
Searching.....	4
Script Generation	6
Connections	8
Exclusions.....	10
Options.....	11
Conclusion.....	12

Introduction

DBS2QVLS will work with any MS-SQL server database that you can define an ODBC connection string for and provides the following functions:

- 1) Search across the entire DB or selected tables in the DB for a particular string or number.
 - a) This comes in very handy when you know a particular string or value is coming out of a DB, but you have no idea what table(s) it is in.
- 2) Quickly give you record counts for all or selected tables within the DB.
- 3) Quickly give you a list of columns within all or selected tables which are NULL 100% of the time.
 - a) This is helpful so when building QV load scripts you don't pull in columns that will never contain data.
- 4) Quickly build a QlikView LOAD and SQL SELECT script for all or selected tables in the DB.
 - a) The script for each table contains the LOAD followed by the SQL SELECT...FROM, formatted with each column on a separate line.
 - b) No WHERE, JOIN, CONCATENATE or other QV clauses are built.
 - c) Columns that are NULL 100% of the time can be excluded entirely or included in the script (but commented out).
- 5) You may define and store unlimited number of ODBC connection strings.

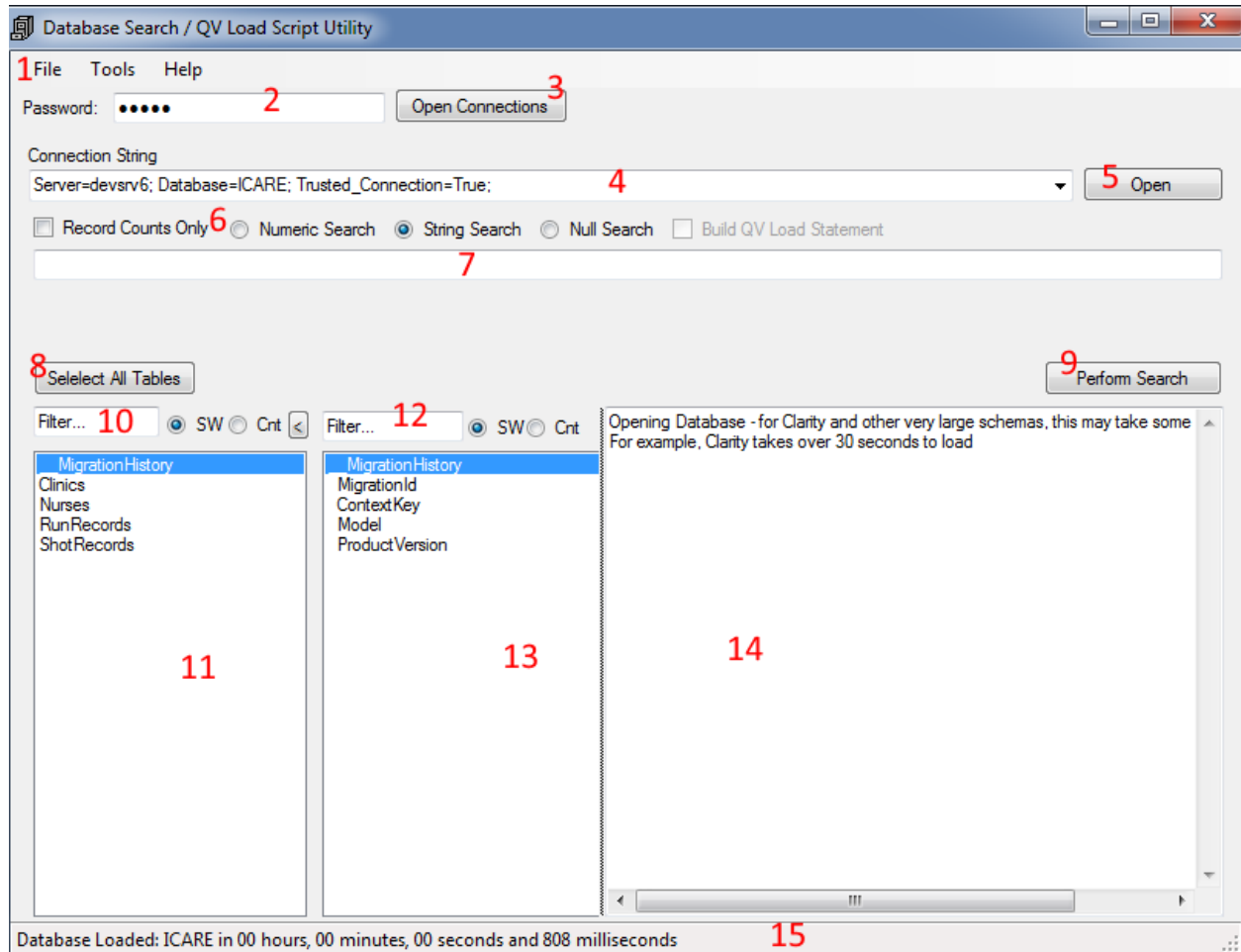
Furthermore, when connected to an Epic Clarity database, the application also supports the following when building QV LOAD scripts:

- 1) Automatically build the requisite MAPPING SELECT ... statements for _C and optionally _ID columns.
- 2) Automatically insert the ApplyMap(...) AS ... statement into the LOAD script for _C and optionally _ID columns.
- 3) For columns with names ending in "_DTTM" or "_DATETIME", optionally add two additional columns containing the DATE (using MakeDate()) and TIME (using MakeTime()).

DBS2QVLS – DataBase Search to QlikView Load Script Utility

- 4) Columns that you do not want to be mapped can be excluded by adding them to the Exclusion list in the application.

Here is the application with a very small database loaded:



A few things to note about this form:

1. The menu contains the following:
 - a. File menu
 - i. Open – will open and unencrypt the defined ODBC connection strings – same as the Open Connections button does – Must enter password first!
 - ii. Exit the application.
 - b. Tools menu
 - i. Exclusions... - Opens the **Exclusions** form.
 - ii. Options – Allows you to view and change available options.
 - iii. Connections - Opens the **Database Connection Maintenance** form to create and store ODBC connection strings.
 - c. Help menu

DBS2QVLS – DataBase Search to QlikView Load Script Utility

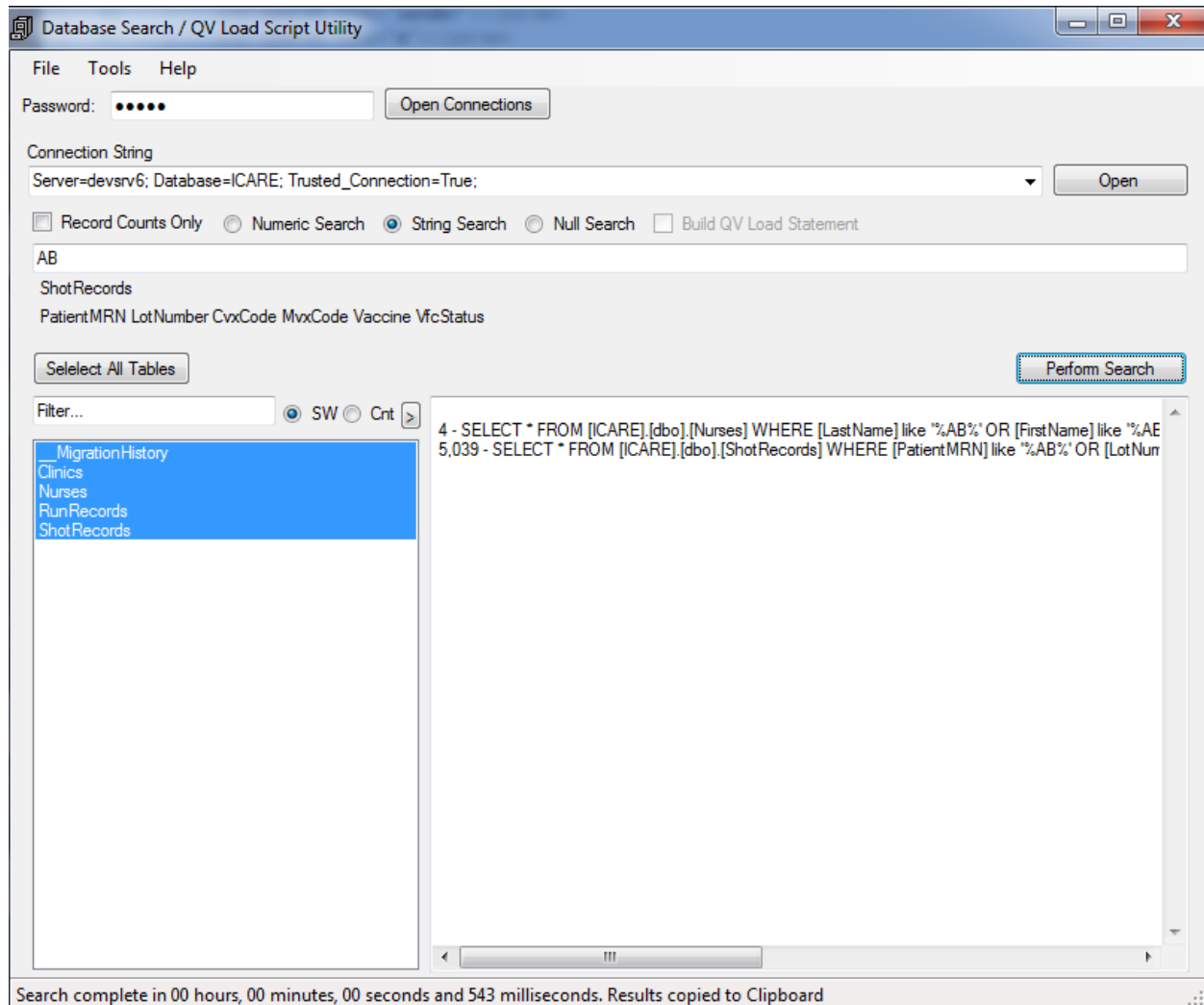
- i. About describes the program.
2. This is the password used to decrypt the file of connection strings you built earlier with the application. Oh, you don't have any connection strings built? After entering a password, click **Tools->Connections** menu item and get busy!
3. **Open Connections** – click this to load your connection strings.
4. **Connection String** – connection strings that have been loaded appear here.
5. **Open** – With an ODBC connection string showing, will open the database; load the tables and column names, finally loading the table names into the ListBox (#11) in the image.
6. **Search Switches** – Control how the application searches.
 - a. **Record Counts Only** – when checked will, for each selected table, return the record count in the table by generating a `SELECT COUNT(*) FROM type SQL SELECT` statement.
 - b. **Numeric Search** – causes the search to occur only on numeric fields within the selected tables.
 - c. **String Search** – causes only CHAR type fields to within the selected tables to be searched.
 - d. **Null Search** – queries each column in the selected tables to determine if the column is NULL 100% of the time. When Null Search is selected, then the **Build QV Load Statement** is available.
 - e. **Build QV Load Statement** – causes the program to build a LOAD-SQL SELECT statement set for the selected table(s) that only includes columns from the tables that are not NULL 100% of the time.
7. **SearchFor** – this is where you enter what you would like to search for in the database. Your entry can be either a string or numeric.
8. **Select All Tables** – Simply selects all the tables in the database for searching.
9. **Perform Search** – Clicking this button performs the search and/or builds the QV statements for the selected table(s).
10. **Filter...** - Enter one or more characters here to filter the list of tables in real-time.
 - a. **SW** – This filters the list of tables to those that START WITH the Filter text.
 - b. **Cnt** – This filters the list of tables to those that CONTAIN the Filter text.
 - c. **>** - This button reveals or hides the column list and has two states:
 - i. **>** - Reveal the column list.
 - ii. **<** - Hide the column list.
11. This list contains all the table names in the database – note only actual tables are included, no views or stored procedures are listed.
12. **Filter...** - Enter one or more characters here to filter the list of columns in real-time.
 - a. **SW** – Filters the list of columns to those that START WITH the Filter text.
 - b. **Cnt** – This filters the list of columns to those that CONTAIN the Filter text.
13. This list contains all the column names in the selected table(s).
14. This text area will contain the results of search operations – also note that results are automatically copied to the clipboard for you.
15. This area is used to keep you updated as to what the application is doing, most of the time.

DBS2QVLS – DataBase Search to QlikView Load Script Utility

Searching

In the list under the **Select All Tables** button, all the table names are loaded and one, several or all of these tables can be selected for searching. In this next image we see the result of searching for the string “AB” in all tables:

Note: The column list is hidden so all columns are searched.



Note in the output area (#12 in the first image), you see the SQL SELECT statements that were generated with a number in front – the number indicates how many rows in the table contain the search string. In the image, the Nurses table contains the string “AB” in 4 rows, and the ShotRecords table contains the string in 5,039 rows. Let’s look closer at the generated SQL SELECT strings, shall we?

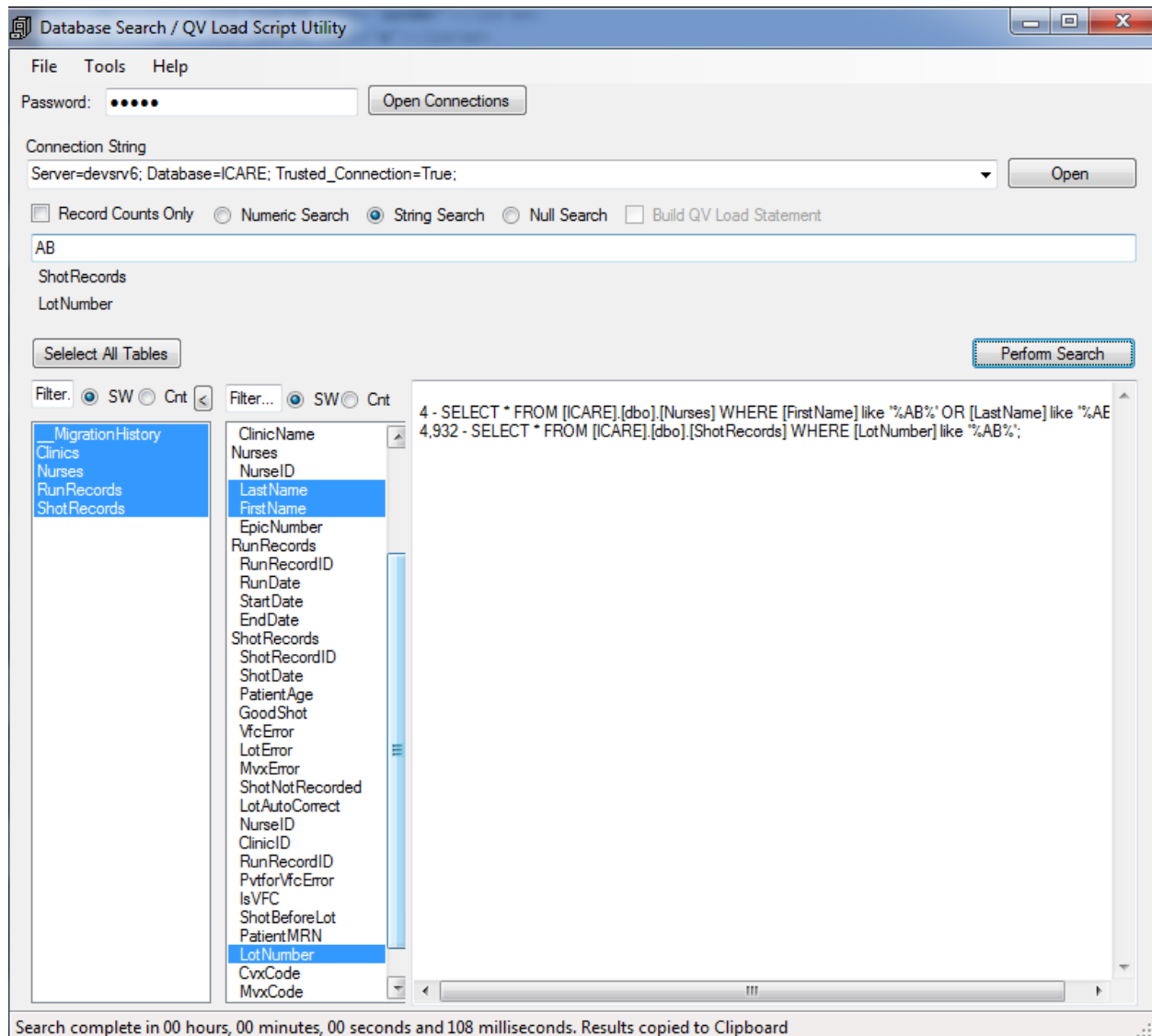
```
4 - SELECT * FROM [ICARE].[dbo].[Nurses] WHERE [LastName] like '%AB%' OR [FirstName] like '%AB%'
OR [EpicNumber] like '%AB%';
```

DBS2QVLS – DataBase Search to QlikView Load Script Utility

5,039 - SELECT * FROM [ICARE].[dbo].[ShotRecords] WHERE [PatientMRN] like '%AB%' OR [LotNumber] like '%AB%' OR [CvxCode] like '%AB%' OR [MvxCode] like '%AB%' OR [Vaccine] like '%AB%' OR [VfcStatus] like '%AB%';

Note only CHAR-type fields are searched – for example, the Nurses table contains an INT field (NurseID) field that is not included. Likewise, ShotRecords contains several INT, BIT and DATETIME fields that are not included.

Furthermore, if the column list is revealed, the SQL generator will only consider selected columns, as can be seen in the next image:



These are the SQL statements created when columns are selected:

4 - SELECT * FROM [ICARE].[dbo].[Nurses] WHERE [FirstName] like '%AB%' OR [LastName] like '%AB%';

QlikView is Copyright © 2017 Qlik. All rights reserved.

Epic and Clarity are Copyright © 2016 Epic Systems Corporation. All rights reserved.

DBS2QVLS – DataBase Search to QlikView Load Script Utility

4,932 - `SELECT * FROM [ICARE].[dbo].[ShotRecords] WHERE [LotNumber] like '%AB%';`

Note that only the selected columns were considered and also note that fewer records were found in ShotRecords because the search was limited to a single column in the table.

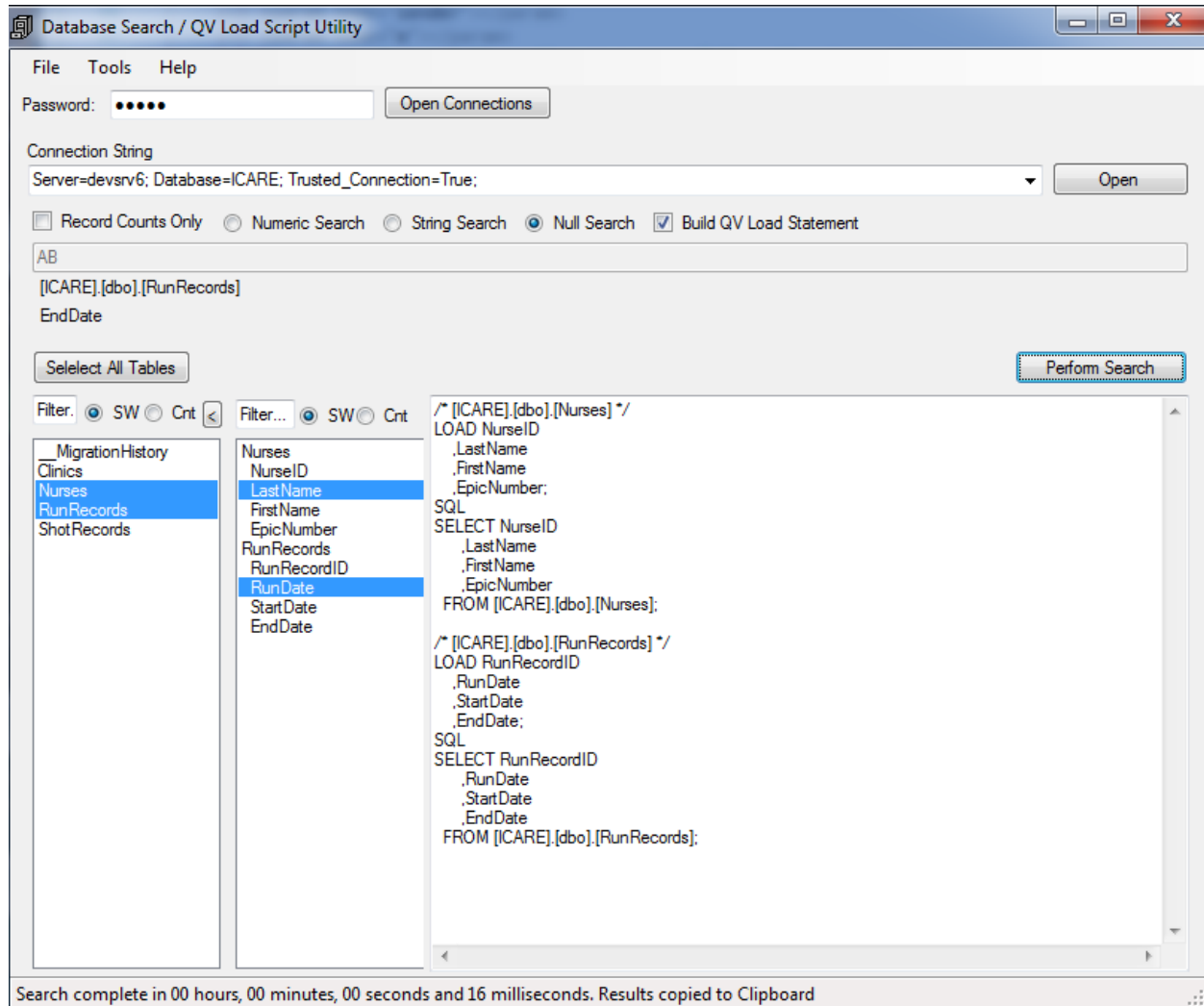
When the SQL SELECT strings are built, the length of the search string is taken into account, and only fields large enough to contain the search string are searched. For example, in the following search string you will note the CvxCode and MvxCode fields are not included – this is because both of those fields are only 3 chars wide.

*15 - `SELECT * FROM [ICARE].[dbo].[ShotRecords] WHERE [PatientMRN] like '%M024261%' OR [LotNumber] like '%M024261%' OR [Vaccine] like '%M024261%' OR [VfcStatus] like '%M024261%';`*

Script Generation

Now let's have some fun, shall we? Choose **Null Search**, then check **Build QV Load Statement** tells the application to build LOAD – SQL SELECT statements that can be dropped into a QlikView Load Script and executed immediately – click **Perform Search** and the following image speaks for itself:

DBS2QVLS – DataBase Search to QlikView Load Script Utility



Note: The column list has no effect on how the QV script is built.

And since the QlikView load statements are copied to the Clipboard for you, you only need to add any necessary “Where” clauses to the pre-built statements, so you can really save some time.

But wait! There’s more!

If you connect to our Epic® Clarity® database, the generated QlikView load statements will also include the necessary Mapping Select and ApplyMap() statements prebuilt for you.

As a QlikView user, you know how powerful ApplyMap() is and you already know there are literally hundreds of lookup fields sprinkled throughout the Clarity® SQL reporting database.

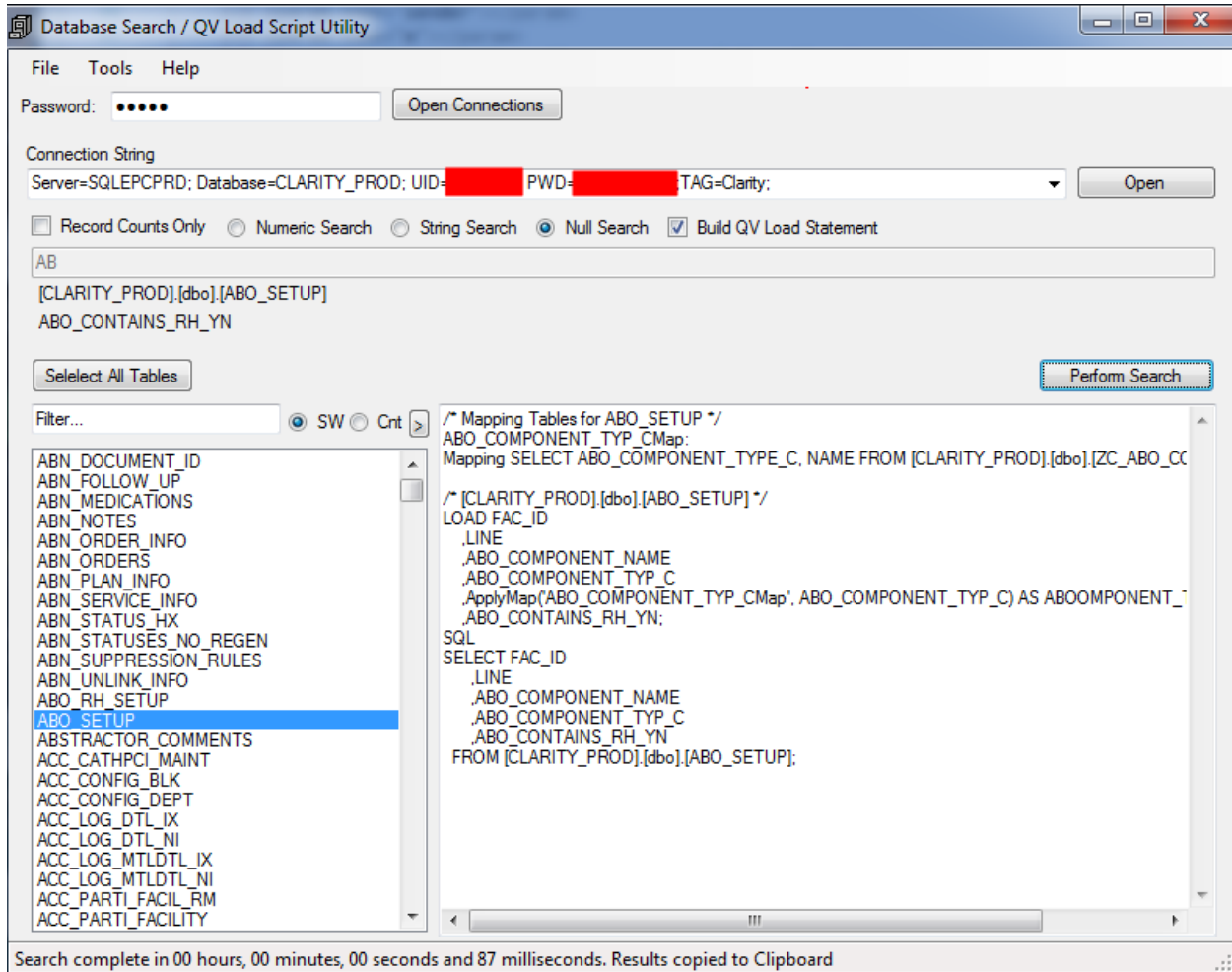
You tell the application you are connected to Epic®’s Clarity® database via an application-specific element on the ODBC connection. When you include “**TAG=Clarity;**” in the ODBC connection string, the application assumes this connection is Epic®’s Clarity® database and enables the ApplyMap() functionality when building the load script.

QlikView is Copyright © 2017 Qlik. All rights reserved.

Epic and Clarity are Copyright © 2016 Epic Systems Corporation. All rights reserved.

DBS2QVLS – DataBase Search to QlikView Load Script Utility

The following image gives you a taste:



Connections

Managing ODBC connections is straight forward in the application, by choosing **Connections** from the **Tools** menu. This opens the **Database Connection Maintenance** form which looks like this:

DBS2QVLS – DataBase Search to QlikView Load Script Utility

Database Connection Maintenance

1 Builder 2 Add 3 Delete 4 Save 5 Close

6

7 Update

8

Server=SQLEPCPRD; Database=CLARITY_PROD; UID=[REDACTED] PWD=[REDACTED] TAG=Clarity;
Server=devsrv6; Database=ICARE; Trusted_Connection=True;
Server=SQLSRVDSK; Database=ServiceDeskPlus; Trusted_Connection=True;
Server=SQLEPCPRD; Database=RHS_REPORTS; UID=[REDACTED] PWD=[REDACTED]
Server=cxprodserver; Database=cx_data; UID=[REDACTED] PWD=[REDACTED]
Server=HBISERVER; Database=MAI_DataCenter; UID=[REDACTED] PWD=[REDACTED]
Server=Sqlpcprddw; Database=CDW_PRD; UID=[REDACTED] PWD=[REDACTED]

Notes on this form:

1. **Builder** Button – Click this to open the ODBC Connection Builder dialog.
2. **Add** Button – Used to add a new connection to the list.
3. **Delete** Button – Click to delete the currently selected connection string from the list.
4. **Save** Button – Click to encrypt and save the list of connections.
5. **Close** Button – Encrypts, saves the connections and closes the form.
6. This field is where you enter and/or edit a new/existing connection string.
7. **Update** Button – Becomes active when typing in the text box (#6) – updates the list when clicked.
8. The defined connection strings will appear here and **ONLY** connection strings listed here are encrypted and saved.

When this form opens and you have connections stored in the Connections.txt file located in the folder with the EXE, they will be decrypted and loaded, as shown above. If you have not defined any connections yet, the form will be empty and you must either type in your connection string or click the **Builder** button to build a connection string and then click the **Add** button to add the connection to the list.

Note: Only when a connection string is displayed in the list will it be encrypted and saved.

Clicking the **Builder** button will open the following form which will help you build a connection string:

DBS2QVLS – DataBase Search to QlikView Load Script Utility

The screenshot shows the 'ODBC Connection Builder' window. It has a title bar with standard Windows controls. The main area is titled 'MS SQL Server Only!'. It contains several input fields and radio buttons, each with a red number annotation: 1 points to the 'Server' text box; 2 points to the 'Database' text box; 3 points to the 'Epic Clarity DB?' radio buttons (Yes/No); 4 points to the 'Trusted Connection' radio buttons (Yes/No); 5 points to the 'User ID' text box; 6 points to the 'Password' text box; 7 points to a large text area below the password field; 8 points to the 'Test' button; 9 points to the 'Accept' button; and 10 points to the 'Cancel' button. Below the password field, there is a note: 'Tab out of Trusted Connection Yes or Password field to build the connection string'.

Notes on this form:

1. **Server** – Enter the name of the SQL server here.
2. **Database** – Enter the name of SQL database here.
3. **Epic Clarity DB?** – Choose Yes if this connection string is for Epic®'s Clarity® database.
4. **Trusted Connection?** – Choose Yes if AD credentials are used, No if you need to specify User ID and Password for this connection.
5. **User ID** – Enter the User ID used for this connection.
6. **Password** – Enter the correct password for this connection.
7. When Trusted Connection is set to “No” or the Password field loses focus, the resulting connection string is displayed here.
8. **Test** – Click this to perform a test connection to the database using the current entries.
9. **Accept** – Click this to close the form and transfer the connection string back to the **Database Connection Maintenance** form.
10. **Cancel** – Click this when you are too chicken to continue – the form closes and you are left with nothing...

Exclusions

When generating QlikView© Load Scripts against the Epic© Clarity© database, the application can generate Mapping Load and ApplyMap() sequences that you may not want it to. The Exclusions form gives a simple way of preventing the application from considering certain fields in the database.

DBS2QVLS – DataBase Search to QlikView Load Script Utility

To access this form, choose **Exclusions...** from the **Tools** menu. The form consists of simply a **Grid** control and a **File** menu.

Simply enter the name of the **Source Table** and the name of the **Source Column**, making sure you TAB out of the **Source Column** field before saving the Exclusions by choosing **Save** from the **File** menu.

When all excluded fields are entered, choose **Exit** from the **File** menu to close the form.

Options

When generating QlikView© Load Scripts against the Epic© Clarity© database, you have some control over what the application generates by setting several options, accessed by choosing **Options** from the **Tools** menu:

Options

Clarity (r) Script Generation Options

Map _C Fields? ☒

Map _ID Fields? ☒

Create Date field from _DTTM fields using MakeDate()? ☒

Create Time field from _DTTM fields using MakeTime()? ☒

Create Date field from _DATETIME fields using MakeDate()? ☒

Create Time field from _DATETIME fields using MakeTime()? ☒

Include Null Fields? ☐

For _C & _ID fields, the ApplyMap Table should be named:
(For example, SEX_C becomes SEX_CMap)

For _C & _ID fields, the Expanded Field should be named:
(For example, SEX_C becomes SEX_NAME)

Accept Apply Close

The options available are described below:

- **Map _C Fields?** – When checked causes the Load Script to include necessary Mapping Load and ApplyMap() code for columns with names ending in “_C”.
- **Map _ID Fields?** - When checked causes the Load Script to include necessary Mapping Load and ApplyMap() code for columns with names ending in “_ID”.

DBS2QVLS – DataBase Search to QlikView Load Script Utility

- **Create Date field from _DTTM fields using MakeDate()? –** When checked, directs the Script generator to build a MakeDate() function to isolate the date portion of a datetime into a separate field. This is for fields that end with “_DTTM”.
- **Create Time field from _DTTM fields using MakeTime()? –** When checked, directs the Script generator to build a MakeTime() function to isolate the time portion of a datetime into a separate field. This is for fields that end with “_DTTM”.
- **Create Date field from _DATETIME fields using MakeDate()? –** When checked, directs the Script generator to build a MakeDate() function to isolate the date portion of a datetime into a separate field. This is for fields that end with “_DATETIME”.
- **Create Time field from _DATETIME fields using MakeTime()? –** When checked, directs the Script generator to build a MakeTime() function to isolate the time portion of a datetime into a separate field. This is for fields that end with “_DATETIME”.
- **Include Null Fields? –** When checked, directs the script to include fields that are always null in the load script as commented out entries. This option works for ALL databases, not simply Clarity©.
- The final two options allow you some measure of control over how the ApplyMap() table is named and what should be appended to the new field created by the mapping.
- **Accept –** Click this to save the options and close the form.
- **Apply –** Click this to save the options and leave the form open.
- **Close –** Click this to close the form without changing the options.

Conclusion

In conclusion, this application has saved me countless hours of digging through undocumented databases searching for specific data in order to build reports and/or QlikView applications. With the added ability to create the majority of a load script for QlikView for any table in any connected database, I save even more time and do way less grunt work building SQL statements to pull data.

There are a lot of ways this application can be improved, and I look forward to seeing where the community take this application in the future.

Enjoy!