



CU2 Global Pty Ltd
The Global Data Conversion Experts



ConvertU2 Technologies
The Data and Software Conversion Experts



2SQL Conversion Project Services Guide

October 2015

© 2015 ConvertU2 Technologies Pty Ltd in conjunction with CU2 Global Pty Ltd. All rights reserved.

ConvertU2, CU2, and 2SQL are either registered trademarks or trademarks of ConvertU2 Technologies Pty Ltd in Australia, the United States and other countries. Access and SQL Server are trademarks of Microsoft Corporation.

Table of Contents

INTRODUCTION	3
SERVICES RELATED CONVERSION CATEGORIES (CLEAN-UP)	4
RUN THE 2SQL GENIE.....	5
POST CONVERSION CLEAN-UP ISSUES.....	5
<i>Clean-up Category: Access Queries.....</i>	<i>6</i>
<i>Clean-up Category: User Defined Function Templates.....</i>	<i>7</i>
<i>Clean-up Categories: DAO Recordset Instantiations, DAO Executes, RunSQL Usage.....</i>	<i>8</i>
<i>Clean-up Category: Calls to SetRowSource.....</i>	<i>8</i>
<i>Clean-up Category: Dynamic RecordSource Objects</i>	<i>8</i>
<i>Clean-up Category: Requery Commands.....</i>	<i>9</i>
<i>Clean-up Category: Identical RecordSource and RowSource Objects</i>	<i>9</i>
<i>Clean-up Category: Unable to Identify Parent Tables</i>	<i>10</i>
<i>Clean-up Category: String based date comparisons</i>	<i>10</i>
<i>Clean-up Category: Wildcards.....</i>	<i>11</i>
<i>Clean-up Category: Microsoft Access Conversion Issues</i>	<i>11</i>
<i>Clean-up Category: Microsoft Access Conversion Issues</i>	<i>12</i>
<i>Clean-up Category: Microsoft Access Conversion Issues</i>	<i>12</i>
<i>Clean-up Category: DDL Commands.....</i>	<i>12</i>
<i>Clean-up Category: String based Boolean Expressions</i>	<i>13</i>
APPLICATION TESTING.	13
DEPLOY TO PRODUCTION ENVIRONMENT.	14

Introduction

In addition to the core 2SQL software CU2 Global provides software utilities and documented processes which support the process of arriving at the desired result – a fully and completely converted database.

This document highlights the 2SQL Conversion Categories where manual services may apply after the 2SQL Genie has completed automatically migrating and converting a Microsoft Access database to SQL Server/Azure.

It is expected that a “developer” undertaking a 2SQL Conversion Project has a comprehensive knowledge of Microsoft Access and SQL Server programming.

This document should also be read in conjunction with the 2SQL User Guide, 2SQL Technical Reference Guide, and the 2SQL Toolkit Document.

Services Related Conversion Categories (Clean-Up)

For the simpler applications, it is possible for 2SQL to deliver a 100% automated solution. For the more complex applications, it is impossible to automate 100% of the Conversion Issues. 2SQL goes very close, in fact in most cases the software automates in excess of 95% detection and remedy of all of the issues. For example, with 2SQL a project that might take a year to do manually can be reduced to less than three weeks of services/cleanup work, and a project that takes months manually should be reduced to days.

What is 'Cleanup' and why is it so important? The salient point is that an incomplete conversion cannot deliver a reliable and robust solution - no stone must be left unturned. And herein lies the attraction of using 2SQL to automate the conversion process and the CU2 knowledge base to complete it. With the 2SQL Detective, the 2SQL Genie, and this Conversion Project Services guide - nothing escapes detection and resolution.

2SQL will report on conversion issues that it can find and remedy, and where possible highlight those it cannot. However, there are other factors besides specific coding issues, and this section discusses them in some depth. The factors highlighted either cannot be automated, or are one of a small group of issues that are outside of the functional scope of 2SQL. It is important to note that the 2SQL Detective is capable of detecting every known conversion issue, and that the 2SQL Genie is capable of well in excess of 95% automatic remedy of them.

It is a fundamental assumption that the staff undertaking the clean-up work have expertise in both the programming and the administration aspects of Microsoft Access and SQL Server.

This next section assumes that your Access database has already been processed by 2SQL in Detective Mode. Refer to the sections discussed in the 2SQL User Guide for more information on how to prepare your database for the 2SQL Detective.

Run the 2SQL Genie.

Data and Software Upsizing

Select the front end Microsoft Access database in 2SQL and process in Genie mode using the same options that were chosen to run the 2SQL Detective. 2SQL will create the SQL Server database, create all the tables, and migrate the data. 2SQL will continue on to convert the SQL Statements in the query, form, and report objects to views or stored procedures, and optimize the VBA code. 2SQL will then create a new Access Front End Database that is linked and synchronized to the new SQL Server Back End Database.

On completion of 2SQL Genie processing, select the database once more in 2SQL and click on the Export Results button. This will create another Microsoft Access database containing the results of the 2SQL Genie, and also very important statistical information about the data and metadata. This is called the **2SQL Audit Information** database.

Following are the steps to be followed post processing of an Access Database by 2SQL in Genie Mode:

Post Conversion Clean-up Issues.

When 2SQL processes a database, it logs every issue detected and remedied against its 2SQL Conversion Category, which in turn belongs to its own 2SQL Conversion Component. These categories and components are explained in the 2SQL Product Overview and 2SQL Technical Reference guide. Some of these categories guarantee that there will be some clean-up work required, if there are issues logged against them. Others may require some clean-up work, depending on the nature and permutation of the particular issue. In addition, there can be a number of issues detected by 2SQL that will require some clean-up work. This section covers all of these types of cleanup issues in their corresponding conversion/cleanup category.

In the 2SQL Audit Information database, there is a table called DBAlerts. This table contains Alert messages of type Design, Warning, Cleanup and Error Exception that the 2SQL Genie encountered when it processed the database. All message types except the Design Type need to be reviewed and addressed manually if necessary. Issues logged with alerts of type Design are issues that are faulty or ambiguous in the original database before conversion, and are logged to assist in the troubleshooting process of the conversion project in general.

Each issue logged into the DBAlerts table also includes a clean-up rating. This is typically the number of minutes a qualified programmer would take to review and resolve the issue at hand. The purpose of this rating is to assist CU2 Global Alliance Partners or CU2 Global 2SQL customers in estimating the amount of work required to complete a conversion project.

The clean-up issues logged in the DBAlerts table can be summarized into the following clean up categories:-

Clean-up Category: Access Queries
Component: SQL Server Objects

Open up the newly created database in SQL Server. Views and Stored Procedures that were not upsized by 2SQL will have been uploaded as stored procedures in a 'commented' state with "from 2SQL" appended to the name. These objects will not have upsized due to either:

1. The original query from Microsoft Access was not in a functional state to begin with, and escaped detection of this by the 2SQLGenie, OR
2. The issues causing the failure to upsize them are currently outside the scope of 2SQL.

Each view or procedure not upsized due to the latter point above will require manual intervention to fix. Note that the comments include both the converted T-SQL command and the original command from Microsoft Access. No more than 5% of SQL Statements processed by 2SQL should be in this 'limbo' state of compilation. Clients are encouraged to advise CU2 Global of the cause so as to keep improving the 2SQL Technology.

Once the issue at hand has been addressed, the following needs to occur:-

1. Execute the CREATE VIEW or CREATE PROCEDURE in the SQL Server Query Window.
2. Delete the corresponding "from 2SQL" object
3. For stored procedures, a pass thru link may also need to be created in the front end. Views and Tables are detected automatically during the re-linking process in the step below.
4. Execute the SQLServerConnectionSettings Macro in the Microsoft Access Front End database, and click the command button to re-link tables, views, and stored procedures. This will create ODBC links for the objects repaired manually, making them available in the front end database.

Clean-up Category: User Defined Function Templates
Component: SQL Server Objects

2SQL will have created a SQL Server User Defined Function Template for each User Defined Visual Basic function that requires conversion. These functions are used by SQL Statements within the application's Views, Stored Procedures or VBA Code. Each template must be completed by manual conversion from the original VBA function. Note that the original VBA code has also been migrated across to SQL Server but in a commented state so as to assist in the manual conversion of VBA Code to T-SQL.

Clean-up Categories: DAO Recordset Instantiations, DAO Executes, RunSQL Usage
Component: Optimisation of Data Objects

2SQL will convert DAO Recordsets to ADO, DAO Executes, and Docmd.RunSQL to their ADO/OLE equivalent, but not the corresponding SQL Statements. The SQL Statements need to be reviewed manually and converted from JET to T-SQL Syntax, if the SQL Statements are not the names of a table or query object in the original database.

A future version of 2SQL will automate the conversion of SQL Statements in the VBA code if they are literal string expressions encapsulated by double quotes, as opposed to VBA variables.

Clean-up Category: Calls to SetRowSource
Component: Optimisation of Data Objects

The 2SQL Toolkit has a method called SetRowSource. This method needs to be implemented whenever the rowsource property of a form or report control is initialized with a SQL Statement in the VBA code. The SQL Statement itself also needs converting to T-SQL Syntax. See the section about the SetRowSource method in the 2SQL Toolkit document for more information.

Clean-up Category: Dynamic RecordSource Objects
Component: Optimisation of Data Objects

The SQL Statements of RecordSource properties of Forms and Reports that are set at runtime in the VBA Code need to be converted to T-SQL Syntax. The 2SQL Toolkit has a method called SetRecordSource. This method can be implemented whenever the RecordSource can be read only. Otherwise the SQL Statement needs to become a VIEW in SQL Server, with the unique index columns added to the tbl_PseudoIndexes table, and the RecordSource property changed to the name of the View. See the section about the SetRecordSource method in the 2SQL Toolkit document for more information.

Clean-up Category: Requery Commands
Component: Optimisation of Data Objects

The VBA code can often instruct a requery of either a RecordSource or RowSource property of Forms. When these properties depend on the values of other form controls, a call to the 2SQL Toolkit function called SetHostValue will also be required. These calls will already be in the gotfocus event of the form control that the requery depends on. The relevant calls need to be copied to the line before the requery.

Clean-up Category: Identical RecordSource and RowSource Objects
Components: Application Framework/SQL Statements/SQL Server Objects

There are occasions when a Form can have the same Query Object or Identical SQL Statement for its RecordSource property and one of the RowSource properties of a control on the same form. Sometimes this can cause the form to timeout, or cause SQL Server Deadlocks. In such instances, change the sql statement of the View attached to the rowsource property to be different to the sql statement of the View that the recordsource is attached to. On most occasions, this will be a matter of creating a new View for the rowsource control from its original and then reducing the number of fields in the SELECT clause of the (new) RowSource View, without compromising the functionality of the rowsource control. 2SQL logs such instances as an Alert for manual review during the conversion process.

Clean-up Category: Unable to Identify Parent Tables
Components: SQL Server Objects, ODBC Connection Objects

2SQL will create INSTEAD OF TRIGGERS on the SQL Server Views that were originally updatable queries with JOIN clauses in Microsoft Access. 2SQL depends on data in the underlying tables to determine the unique keys for these triggers, and also the ODBC Pseudo Index. If there is no data in the underlying tables then 2SQL is unable to do this. In such cases the triggers and the pseudo index need to be implemented manually.

Clean-up Category: String based date comparisons
Components: SQL Server Objects. SQL Statements

When the FORMAT function is used to format a Date Data Type, it will return a String Data Type, not a Date Data Type. However, several Microsoft Access developers are not aware of this, and as such there can be occurrences of comparing a string based date expression, to a Date Data Type, in the SQL Statements of Queries and VBA Modules . For example, `FORMAT(datefield, "dd/mm/yyyy") >= NOW()`. The JET Engine in Microsoft Access appears to be built to consider this oversight by developers, and when executing the SQL, it will implicitly convert the string based date to a Date Data Type, before it does the comparison. With SQL Server, it is the other way around. SQL Server will implicitly convert the Date Data Type to a String, if it is being compared to a String data type. As such, discrepancies can occur between Access and SQL Server, for their corresponding SQL Statements. 2SQL will generate a warning log for occurrences of String based Date Comparisons. These occurrences need to use the SQL Server DateDiff Function as the means for comparison.

Clean-up Category: Wildcards
Components: Jet SQL Statements

Sometimes it is more appropriate to use a JET based SQL Statement in a Form RowSource or RecordSource, or in the VBA code. If such statements use WildCards as part of the LIKE predicate, ensure that the SQL Server compatible Syntax option is turned off, in Microsoft Access, and use ANSI 89 Wildcard Syntax, not ANSI 92 Syntax. Using ANSI 92 syntax against tables that are linked to SQL Server can produce incorrect results.

Clean-up Category: Microsoft Access Conversion Issues
Issue: Form Properties

It is not uncommon where a 2SQL Conversion Project also involves converting the Microsoft Access database from an earlier version (eg Access 2000 to Access 2010). 2SQL will programmatically convert Access databases to the version of its own Installer (2010 or 2013), using the built in ConvertAccessProject VBA command, which is the equivalent of "Convert Database...", in the Access User Interface. However there are bugs in this conversion utility and such bugs can compromise the converted outcome, depending on the content of the Access database.

The area of potential discrepancy is in Microsoft Access forms. There are several properties that are new, and did not exist in earlier versions. For example, the "DataEntry", and "OrderByOnload" properties of a Microsoft Access form. Sometimes the conversion process will set these values incorrectly. If behavior is not correct after conversion, check the form properties and ensure they are correctly set. Sometimes different behavior also occurs not because the settings are different, but because Microsoft have done some bug fixes, which makes the setting work where it did not before. The setting will need to be changed for desired effect, if behavior is not the same as the original database.

Clean-up Category: Microsoft Access Conversion Issues
Issue: Form Properties – Enter Key Behaviour

If the original version of the Access database is earlier than 2010, there may be some occurrences where pressing the Enter key in the text box of a form, will force navigation of the form to go the next record. If this occurs whereas it did not before the conversion, change the "Enter Key behaviour" of the text box, to "New Line in Field".

Clean-up Category: Microsoft Access Conversion Issues
Issue: Form Current Event

The Form Current event may be firing in a Microsoft Access form where it did not in the original version. This is because of bugs fixed by Microsoft where a particular setting that was originally dormant because of the bug, is now active and recognized. The VBA Code content of the Form Current event may need modification if behaviour is different or not the desired outcome.

Clean-up Category: DDL Commands
Component: SQL Statements

SQL Statements that are DDL (Data Definition Language) based, are not converted by 2SQL to their SQL Server equivalents. DDL statements (eg CREATE TABLE....) are typically very low in their occurrence, and very often they are no longer required. If they are required, they must be identified and converted manually.

Clean-up Category: String based Boolean Expressions
Component: SQL Statements

2SQL will convert the JET built in Boolean Expression Types such as Yes/No, True/False, to 1 or 0 respectively, in the SQL Statements that are converted to SQL Server. JET also has the capability of dynamically converting String based Boolean Expressions, into a data type of Boolean, for comparison purposes of True or False.

Occurrences of String Based Boolean Expressions must be converted manually, typically by using a CASE statement. For example:-

WHERE fieldname = True

Becomes

WHERE (CASE WHEN fieldname = 'Yes' THEN 1 ELSE 0 END) = 1

Application Testing.

Once the clean-up issues above have been checked and implemented where necessary, the ODBC Refresh Interval for Microsoft Access needs to be checked if the Application provides data input via Microsoft Access Forms. This is highly likely of course, in which case it is best to set the ODBC Refresh Interval to 2 (Two) Seconds. This setting can be maintained directly from within Microsoft Access via the Tools>Options>Advanced in Microsoft Access 2003, or Office Icon>Access Options>Advanced in Microsoft Access 2007 and 2010.

The application is then ready for testing. General testing should occur first wherever possible, followed by joint testing with a representative who has a very intimate understanding of how the application is supposed to behave and function.

Deploy to Production Environment.

Upon completion of User Acceptance Testing, it will be time to rollout the solution for the end users. 2SQL does this automatically via the Deploy button on the main form. This feature of 2SQL should be practiced for as many times as need until perfect during the services side of the conversion project. See the section about Deployment in the 2SQL User Guide for more information.

Disclaimer

The Access to SQL conversion and migration challenge is a very complex subject. Whilst every effort has been made to completely and accurately represent the conversion issues and their corresponding solution, we do not claim perfection in our analysis. For that reason we welcome reader feedback to info@cu2global.com as part of our striving to continue to make 2SQL the best product of its type in the world market today.
