CU2 Global Pty Ltd

The Global Data Conversion Experts

ConvertU2 Technologies

The Data and Software Conversion Experts

White Paper

Converting Microsoft Access to SQL Server

October 2013

Introduction

Microsoft Access is the world’s most popular, flexible and forgiving database development platform: semi-skilled developers can create a wide variety of very useful applications and skilled developers are able to develop extremely functional and complex applications, many of which are mission-critical.

However, Access has a number of significant inbuilt limitations whichin many ways Microsoft has itself acknowledged by equipping the Access 2013 release with an SQL Server back-end.

The purpose of this White Paper is to highlight some of those limitations and discuss ways in which older versions of Access, using desktop technology, can be converted to the much more powerful SQL server-based technology. This change will enable an organization to stay current with the latest releases of Microsoft Office. In many businesses that step alone can be mission-critical because not staying current with software releases can be a serious roadblock to integration with other modules in the Office suite

What are the Benefits of converting from Access to SQL Server?

Discussions on the subject of Access versus SQL Server are usually provoked by one or more users of the more important Access Applications bumping into any number of the Access limitations listed in the table below.

The more significant key drivers that stimulate consideration of the various conversion options are:

- corporate governance surrounding data security and integrity,
- data collaboration (sharing),
- and the need to keep Access current with new releases of Microsoft Office.

A further example is the necessity to spend more on hardware and software to beef up a network whichcan become very slow if it is frequently overloaded with a large number of queries working with large file sizes that are often involved in the reporting process.

	ISSUE	ACCESS LIMITATIONS	SQL SERVER CAPABILITY
1.	**Data and Software Scalability**	Has numerous limitations	Almost infinitely scalable in every respect.
2.	**Business Intelligence**	Non-existent	Major core functionality
3.	**Data repository**	Resides on the desktop or network drive, posing a threat to data security via access from a forbidden user; conversely blocking access from someone who needs to use it.	Resides on a centralised server, making data much easier to access, manage and maintain.
4.	**Data/software security and permissions levels**	Password protected and encrypted	Much stronger security levels and versatility and a must if security and permission levels are critical
5.	**Data integrity, backup and maintenance**	Backup and recovery is not robust, with a chance of failure. Cannot be backed up if the file is open and the data changing – a major problem with critical 24/7 applications	Are robust core functions. Has a robust set of administration tools including detailed transaction logs which report on which user effected modifications and to what data, and has rollbacks to undo changes
6.	**Maximum database size**	Can theoretically support up to 2GB – practical experience shows that the larger the database, the less robust Access becomes	Is a far better choice for managing large database files

	ISSUE	ACCESS LIMITATIONS	SQL SERVER CAPABILITY
7.	**Number of concurrent users**	Has problems with too many concurrent users, the number depending on how well the application was designed and meant to be used	Supports more users and more network traffic, with opportunity to improve performance by upgrading server hardware. This option is not leveraged as much for file server databases such as Access
8.	**Performance**	Processing is much slower when it occurs on distributed desktop computers	A centralised SQL Server computer will process Access data much more quickly than desktop computers
9.	**Network bandwidth**	Access databases pass the whole table (or at least the index) across a network to fulfil Query reporting. Large files can 'choke' a network and badly affect performance	An optimised Application using SQL Server significantly improves network performance because only selected records are passed from the database to the Application
10.	**Data collaboration**	Data residing in Access on desktops may not be accessible by other desktops on the same Local Area Network	The SQL Server platform is designed to facilitate full data collaboration across a network
11.	**Software development environment**	Both Access and SQL Server have software development features exclusive to themselves	The SQL Server development toolset provides far more leverage, e.g., asynchronous processing can be introduced by way of SQL Server agent jobs. There are many more additional facilities
12.	**Software re-engineering**	Access Applications often become convoluted and inefficient in their design. Complex Applications push the limits of the technology, compromising robustness and the ability to cost-effectively maintain and further develop	The SQL Server environment provides a much wider-ranging toolset for software re-engineering
13.	**Web enablement**	Although an ASP.NET solution can use Access as a data source, it is neither robust nor reliable	ASP.NET was built around SQL Server and reports can be efficiently converted to web-enablement using SQL Server Reporting Services
14.	**Azure Cloud hosting**	Pre 2013 versions of Access are not supported by the Microsoft Windows Azure software	To take advantage of Microsoft SQL Azure for the Cloud, Access Application data either has to be converted to SQL Azure compliance or developed specifically for that technology

Upsizing Microsoft Access to Microsoft SQL Server

‘Upsizing’ is Microsoft parlance for upgrading from the **JET** (*Joint Engine Technology*) client-side (desktop) processing to the latest technology **SQL Server**with server-side processing where central servers manage the data much more efficiently than is possible in distributed desktop populations.

In most Enterprises with a dependency on one or more mission-critical Access database Applications, upsizing is a very desirable step. However, the conversion and migration is often a very complex process with a high level of associated risk unless automation is utilised.

Web research will turn up a significant volume of information about Access to SQL Server upsizing in a variety of White Papers, technical notes, Microsoft sites, blogs and even books that have devoted chapters to what is a much-researched subject; everyone is looking for the silver bullet which will provide a complete solution. Until now, there doesn’t seem to have been a product available that would deliver a close-to-100% automated conversion from Access to SQL Server.

What are the main Upsizing options?

Manual Conversion

For Databases of any significant size and/or complexity, this option is not feasible unless the budget to do so is significant and there are no pressing time constraints. Just finding the many thousands of issues to be converted is a huge task on its own, but when found the issues have to be programmatically remedied and the outcomes thoroughly tested.

Testing programmatically produced remedies is the most difficult step in the process because the methodology used to locate and remedy all issues must be bordering on perfect or the whole Application could well produce unreliable outcomes.

Utilising the free Microsoft Access Upsizing Wizard and SSMA (SQL Server Migration Assistant) tools

The Upsizing Wizard provides what is known as a ‘linked tables’ solution but has been superseded by SSMA for Access which has greater functionality. However, the scope of both tools is limited. In short, neither of the tools will fully identify the numerous issues to be remedied and there will still be many manual conversion steps required from where the tools have left off.

The objective of an SSMA conversion is to alleviate some of the shortfalls of Access application solutions and/or to gain some of the benefits of the SQL Server platform. The result is a “hybrid” application with an Access database front end, “linked” to the

SQL back end but still requires considerable manual work on the table data and schema just to get the upsizing to function.

Considerable additional manual work is required in the corresponding software that resides in the Access Queries, Forms, Reports, Macros and VBA Code Modules.

There are several reasons why the Microsoft upsizing tools usually fall short of the ideal:

- Utilising SSMA plus manual programming could deliver a fully functioning Access database with an SQL Server back-end container but the likelihood of a satisfactory outcome decreases in direct proportion to the complexity of the Application being converted.
- Although all table data is stored in SQL Server, Access still uses the JET database engine rather than SQL Server to run queries, store object definitions, manage temporary tables and hold security settings. Only the table schema and data are migrated reliably.
- A knowledgeable developer can convert additional objects that SSMA does not attempt to upsize, but the outcome, whilst more robust, is still an Access Application on the front end running JET with a SQL back end running the engine. The interaction of the two engines can sometimes produce results that are inferior to those obtained by using only JET with native Office Access tables.

The Microsoft Upsizing Wizard and SSMA only provide a fraction of the critical changes to the Access Application, thereby producing what can only be regarded as an incomplete conversion.

Upsizing completely using the 2SQL automated conversion toolset

A complete upsizing has to deliver an Application that takes full advantage of the functionality and power of the SQL Server platform including optimized performance, data security and integrity, scalability through increased database size and number of concurrent users, professional level maintenance and backup procedures.

For a more detailed discussion on this topic, please refer to the 2SQL Functionality Overview.

The choice of making the schema of the SQL Server tables compliant with the Access Services Engine of SharePoint 2013 should be available as an option prior to conversion, even if the solution will not or cannot immediately utilize the advantages of the Access Services model. This is because locally based forms and reports are no longer part of the long-term future of Microsoft Access. Choosing a solution compliant with Access Services creates fewer restrictions for new development.

If an Application is converted to this extent it will be regarded as “SQL Server compliant”. The Access JET database engine will have been reduced to a connection object wherever possible and all ‘processing’ will take place on the back-end SQL Server database.

Depending on the complexity of the Access databases to be converted, ConvertU2's flagship technology, 2SQL, automates 95% to 100% of the effort required to completely upsize a Microsoft Access Application to SQL Server.

It is available via a registration and download process that lets you scan your Applications free of charge using the 2SQL Detective to identify conversion and migration issues.

Upon purchase of an official 2SQL Software Licence, these Applications can then be reprocessed in the Genie (Convert and Migrate) mode by 2SQL, which will remedy 95%-100% of issuesidentified by the 2SQL Detective.

Extensive journaling allows those issues not automatically remedied to be identified for resolution in the post conversion phase.

Post Conversion

There is often a desire to replace the Microsoft Front End completely with a thick VB.NET client or ASP.NET thin client solution.What is not realised or appreciated, however, is that a pre-cursor 2SQL-converted front end will take the developer around 50% of the way to building the required thick client Web-enabled front end in its place.

When an Access front end is converted to SQL Server compliance, all the SQL Statements that reside in the Query, Form, Report and Module objects will then reside in SQL Server as Views or Stored Procedures. This is also a requirement for replacing the front end and is a very large component of such projects.

Performing a conversion of the Access front end first to SQL Server compliance by way of linked ODBC and OLE objects provides an effective way to replace the front end in two manageable phases with very little overlap compared to one ‘do it all at once’ project. In fact, it simplifies the entire process because troubleshooting during the testing phase can be broken down into conversion or re-engineering.

Microsoft Direction

The development landscape for Microsoft Access is much more focused on SQL Server as a data repository with the advent of SharePoint 2013, Office 2013 and Office 365. The Access Services component of SharePoint 2013 is much more versatile in its functionality than SharePoint 2010, and relies exclusively on the data being stored in a SQL Server 2012 or SQL Azure back end.

For new databases, this means that Microsoft Access front ends can be developed to utilize the built in (Access) web page designers for data input, instead of Microsoft Access Forms; and Data Macros instead of VBA code to implement the programmatic component.

For existing databases, however, neither the upsizing Wizard nor the SSMA tools will prepare or ‘upsize’ them for compliance with Access Services. This is because SharePoint relies on various schema rules/conditions of SQL Server thatare out of scope for the upsizing tools.

With 2SQL it is also possible to convert and migrate Access Databases to be compliant with the Access Services engine of SharePoint without compromising any of the existing functionality. This effectively provides the best of both worlds for future development, whereby the existing Forms and Reports can function as they always did using the 2SQL Methodology and future development can be applied in the new landscape of Microsoft Access Web Pages.

Conclusion

With its close-to-perfect automated conversion functionality and ability to comply with the future Microsoft technical direction for its Office suite, ConvertU2’s automated 2SQL product family not only takes you from the past to the present, but also future proofs your investment in upsizing from Access to SQL Server.

References

http://www.cu2global.com/cu2-uni/cu2-global-university