10+ tips for upsizing an Access database to SQL Server

By Susan Harkins July 31, 2008, 8:03 AM PDT

Takeaway: When the time comes to migrate your Access database to SQL Server, you could be in for a bumpy ride. These suggestions will help you sidestep problems — or at least wrestle them into submission.

When the time comes to migrate your Access database to SQL Server, you could be in for a bumpy ride. These suggestions will help you sidestep problems — or at least wrestle them into submission.

There are a number of reasons to upsize an Access database to SQL Server, but the process requires a bit of coaxing. The truth is, upsizing can be a real pain. Access and SQL Server are similar, but they’re not compatible. The good news is that you can take steps to limit problems. At the very least, you should be able to identify problems so you can resolve them.

Note: This information is also available as a PDF download(http://downloads.techrepublic.com.com/abstract.aspx?docid=378663).

#1: If at first you don’t succeed...

Regardless of how well you prepare for the transition from Access to SQL Server, there’s no silver bullet for the process. If you’re lucky, the data will migrate without any problems. Most likely, the data won’t transfer cleanly the first time, or even the second or third time. Don’t be frustrated. Instead, figure out why the data didn’t transfer, fix the problem, and try again.

#2: The SQL Server Migration Assistant Wizard for Access (SSMA)

Access comes with an Upsizing Wizard. To launch it, open the Access database and choose Database Utilities from the Tools menu. Then, select Upsizing Wizard. The process isn’t painless, but the wizard will hold your hand while it hurts.


#3: Mapping data types

It’s a good idea to know what SQL Server will do with your Access data types. During the conversion process, SSMA handles data type incompatibilities for you (for the most part). Table A lists the default type mapping for migrating from Jet 3.5 or Jet 4.0 to SQL Server 2005. Most of them will be exactly what you want, but there might be a few that you have to redefine, which SSMA allows you to do.
### Table A: Data type mapping

<table>
<thead>
<tr>
<th>Jet 3.5 and 4.0</th>
<th>SQL Server 2005</th>
</tr>
</thead>
<tbody>
<tr>
<td>TEXT(size)</td>
<td>nvarchar(size) varchar(size)*</td>
</tr>
<tr>
<td>MEMO</td>
<td>nvarchar(max) varchar(max)*</td>
</tr>
<tr>
<td>BYTE</td>
<td>tinyint</td>
</tr>
<tr>
<td>INTEGER</td>
<td>smallint</td>
</tr>
<tr>
<td>LONG INTEGER</td>
<td>integer</td>
</tr>
<tr>
<td>SINGLE</td>
<td>real</td>
</tr>
<tr>
<td>DOUBLE</td>
<td>float</td>
</tr>
<tr>
<td>DECIMAL</td>
<td>float</td>
</tr>
<tr>
<td>GUID</td>
<td>uniqueidentifier</td>
</tr>
<tr>
<td>DATE</td>
<td>datetime</td>
</tr>
<tr>
<td>CURRENCY</td>
<td>money</td>
</tr>
<tr>
<td>YES/NO</td>
<td>bit, tinyint</td>
</tr>
<tr>
<td>LONGBINARY</td>
<td>varbinary(max)</td>
</tr>
<tr>
<td>BINARY(size)</td>
<td>varbinary(size)</td>
</tr>
</tbody>
</table>

*Jet 3.5

### #4: Dealing with index incompatibilities

For the most part, SSMA converts indexes successfully, but there are a few exceptions:
SQL Server’s nvarchar and varchar columns don’t support indexing. If an Access MEMO field has an index, SSMA will ignore it. You’ll just have to do without it in SQL Server.

The SSMA will add an identity column to any table that doesn’t have at least one unique index or a primary key. The Upsizing Wizard that comes with Access may fail to upsize the table at all. If this happens, simply add an AutoNumber field to the table before upsizing.

SQL Server doesn’t allow multiple NULL values in a column with a unique index. Any Access table with multiple NULL values with a unique index won’t upsize. There’s no workaround for this, short of removing the unique index or the multiple NULL values from the Access table before upsizing.

#5: Preparing expressions

Expressions create special problems for upsizing because a lot of characters and practices simply aren’t compatible. SSMA does its best but often fails. You can help by preparing expressions as follows:

- Replace double quotation mark characters (”) with single quotation characters (’).
- Double embedded single quotation characters. For instance, “O’Hara” becomes ‘O”Hara’ — the middle ” is two single quotation characters, not one double quotation character.
- Replace the pound sign date delimiter character (#) with single quotation characters.
- Replace the ampersand character (&) with the + character for concatenation.

#6: Preparing object names

Access lets you use space characters in object names. SQL Server will accommodate these names as long as you enclose each instance in square brackets. For instance, Last Name must become [Last Name]. Access actually takes care of most of this for you. SQL statements are the one area you’ll have to clean up before upsizing.

SQL Server can handle Access objects that are named using SQL Server keywords or that include special characters if you enclose the object name in quotation marks. Unless you know all the keywords, this one can be hard to predict and manage. SSMA encloses these names in square brackets ([ ]).

You can easily avoid this problem by avoiding space and other special characters and keywords when naming Access objects.

#7: Dealing with Required property in primary key columns

An Access primary key can include columns where the Required setting is set to False — meaning that the field could remain empty. SQL Server doesn’t support this configuration. SSMA usurps the Access setting by assigning the NOT NULL attribute to all columns in the primary key. However, you can avert this behavior as follows:

- Choose not to upsize the table.
- Upsize the table as is, despite the incompatibility.

I don’t have a recommendation on this. Just be sure you have a good reason for ignoring SQL Server’s requirements. You will have to deal with it eventually.

#8: Dealing with unsupported dates

SQL Server’s datetime data type is more limited than Access’. SQL Server supports dates from January 1, 1753, to December 31, 9999. Access supports dates from January 1, 100, to December 31, 9999. SSMA migrates Access tables with out-of-bound dates, but it converts those unsupported dates to its nearest limit. (The Upsizing Wizard won’t upsize the table at all.) There’s no work around short of deleting unsupported dates or storing dates as text values and using complex expressions to manipulate them as date values.

To find unsupported dates in Access before upsizing, run a query using the following expression:

date < #1-1-1753# OR date > #12-31-9999#
#9: What to do with all that SQL?

The SSMA converts SQL statements into Transact-SQL (T-SQL) views. What it doesn’t convert, you’ll have to rewrite. There are at least two SQL incompatibilities you should know about:

- T-SQL doesn’t have an equivalent for Jet SQL’s DistinctRow.
- T-SQL doesn’t support an alias in the same statement in which it’s defined.

#10: What’s this?

Not everything converts as you might expect. In other words, you might not find properties, expressions, or attributes where you expect them. The following list will help you find things that seem misplaced at first:

- If the Allow Zero Length property is set to False in a TEXT or MEMO column, SSMA applies it to a check constraint on the SQL Server column.
- SQL Server doesn’t support a key link between a TEXT and MEMO columns of different sizes. SQL Server maps both columns to the largest type to compensate.
- SQL Server enters a 0 value for any NULL values in an Access Yes/No field.
- MMSA converts a Hyperlink column to a text column, which means they lose their functionality.
- Validation Rules are converted to table constraints.

#11: Where’d it go?

The SSMA does what it can with incompatibilities. However, there are a few things that just don’t translate, as there’s nothing in SQL Server to help. When this happens, you’ll have to rethink your strategy or write a stored procedure. You’ll find no SQL Server equivalent for the following:

- Format() function
- StrConv() function
- Switch() function
- Choose() function
- Domain aggregate functions
- Val() function
- DateValue() and TimeValue functions
- Time() and TimeSerial() functions
- WeekDayName() function
- DateAddW() function
- First() and Last() function

#12: Outwitting stubborn tables

If SSMA can’t upsize a table and you can’t determine the reason, use SQL Server’s Management Studio to import the table. You might get lucky. If not, Management Studio will probably give you more direct and helpful information regarding its failure.

#13: Dealing with lots of data

Transferring hundreds of thousands of records to SQL Server can take a long time. If that becomes a problem, empty the Access tables and upsize the shell. Then, use queries or copy the Access data into the newly upsized SQL Server database.

http://www.techrepublic.com/blog/10things/10-tips-for-upsizing-an-access-database-to-sql-ser...
People who read this...

- 10+ reasons why IT pros hate Microsoft Access (but really shouldn't)(http://blogs.techrepublic.com.com/10things/?p=386&tag=rbxccnbtr1)
- 10+ reasons to use Access (and a few reasons not to)(http://blogs.techrepublic.com.com/10things/?p=401&tag=rbxccnbtr1)
- 10 ways to prevent Access database corruption(http://blogs.techrepublic.com.com/10things/?p=331&tag=rbxccnbtr1)

About Susan Harkins

Susan Sales Harkins is an IT consultant, specializing in desktop solutions. Previously, she was editor in chief for The Cobb Group, the world's largest publisher of technical journals.