

SQL Server 2005 backward compatibility issues to consider

It's always good to be on top of the latest technology, and many DBAs with [SQL Server 2000](#) or 7.0 setups are eyeing a migration to SQL Server 2005. The same DBAs typically have to rework SQL server applications to operate correctly with SQL Server 2005, such as custom databases, in-house front-end programs and others.

Here's a rundown of the most important SQL Server 2005 upgrade compatibility issues for consideration -- the things most likely to immediately break an application if it's switched. (For the sake of this article, an ["application"](#) is anything that makes use of SQL Server -- whether it's a product that contains a SQL Server database or just a front end that passes commands.)

System tables

One of the most important backward-compatibility issues involves the elimination of system tables from SQL Server 2005. If you have existing SQL Server applications that reference system tables directly, you will have to update them.

The good news is that SQL Server 2005 does provide backward compatibility for such functions through compatibility views. Adam Machanic, who is also a contributor to SearchSQLServer.com, wrote a tip that details [how compatibility views work](#). It's worth reading if you are planning to port SQL Server applications that use system table references regularly. (Side note: The sample databases, Northwind and Pubs, are no longer included with SQL Server 2005, so don't be surprised if you don't see them either.)

Deprecated functions

Many system functions and stored procedures supported in earlier versions of SQL Server will not work. Microsoft's Books Online for SQL Server 2005 has an index of all backward-compatibility issues, including database engine changes and deprecated commands. Commands that were marked as deprecated prior to SQL Server 2000 should not have been used in SQL Server 2000 anyway.

Data access technologies

This is a big concern for front-end applications. Microsoft no longer supports many of the older data-access technologies allowed in earlier versions of SQL Server. DB-Library (not updated since SQL Server 6.5), Embedded SQL (E-SQL), Data Access Objects (DAO) and Remote Data Objects (RDO) are all being abandoned. If you have applications that use any of these, you'll need to use the ADO.NET libraries to make them work natively with SQL Server 2005.

One key tool you can have at your disposal is the [Upgrade Advisor](#), a free program available on both the SQL Server 2005 CD-ROM or from Microsoft's site. The Advisor scans your existing SQL Server 7.0 or 2000 setup and produces a report of possible issues that might arise if you upgrade to SQL Server 2005. It does not analyze standalone applications, just the installation of SQL Server itself. It's helpful for

eliminating some of the most pressing issues first. Microsoft also has a moderated [SQL Server Setup and Upgrade forum](#) where many common issues (including application compatibility and deprecated functions) are addressed.

Another useful add-on is SQL Server 2005 Backward Compatibility Components package, available through the Feature Pack for SQL Server 2005. This pack includes SQL Server 2005-updated versions of items used in SQL Server 2000: the Data Transformation Services 2000 runtime (DTS), SQL Distributed Management Objects (SQL-DMO), Decision Support Objects (DSO) and SQL Virtual Device Interface (SQLVDI). Grab this if you have applications that rely on any of these components.