Upgrading to SQL Server 2005: Six migration tips

Upgrading to a new version of SQL Server isn't like patching a desktop application or even moving to a new version of Microsoft Office. It has broad-reaching consequences and can affect the behavior of other applications that talk to SQL Server: data-driven Web sites, reporting tools and so on. Here are six things to bear in mind as you're planning and executing the upgrade to SQL Server 2005:

1. **Get to know SQL Server 2005 first.** Don't make the actual migration process your first hands-on experience with SQL Server 2005. Many things have changed. For instance, Microsoft did away with the Query Analyzer, and instead of the MMC-style Enterprise Manager, there's now the Management Studio. If you don't want to install a full instance of SQL Server 2005, you can download a free virtual machine implementation of a SQL Server 2005 instance or pick up a free copy of SQL Server 2005 Express Edition, whose biggest limitation is any single database that's larger than 4 GB. Otherwise, functionally, they are very similar. One way or another, get used to what you'll be working with long before you actually have to get your hands dirty with the real thing.

2. **Be mindful of your last known good backups before migrating.** It seems terribly obvious, doesn't it? Yet, many people don't make a manual backup of their database schema and data before attempting an upgrade, and instead rely on whatever they have from the last automatic backup. I say "last known good backup" specifically because there are many people who take for granted that the last backup set they created works fine -- without actually testing that hypothesis. This warning goes double if you're using a backup set to restore data on the new instance of SQL Server, and it goes hand in hand with the next item on this list.

3. **Make an effort to migrate in parallel.** Instead of backing up one SQL Server, decommissioning it and installing another, try very hard to have both running side by side, either on different machines or the same machine. Multiple instances of SQL Server can coexist on the same computer if you need them to. You might have to set the memory requirements fairly low for one of them.

   If you're just getting started with SQL Server 2005 and don't actually have it in production yet -- but the point here is not to have both of them running full-bore -- just find a way to get them to coexist long enough to move the data over. One of the nice things about SQL Server is that it's designed to be installed in multiple instances on the same machine, so all you need to do is make sure each instance is named separately during the setup process. Also, a parallel migration means you can transfer data either by restoring a backup made by one instance of the server or by doing a server-to-server transfer. Migrating in parallel makes it that much easier to roll back the migration if things don't work out.

4. **Pay attention to the advice from the Upgrade Advisor.** Grab a copy of the Microsoft SQL Server 2005 Upgrade Advisor. Run it against your current SQL Server 7.0 or 2000 installations and think about the advice it gives you. It's not a big download, and you can run it totally non-destructively -- all it does is analyze your existing setup. SQL Server 2005 site expert Adam Machanic has written his own article about it: SQL Server 2005 Upgrade Adviser reduces unknowns. One of the best things about this tool is that it gives you both problems *and* solutions. You'd want to know if there's a "time bomb" issue sitting around in the database while you're planning to upgrade.

   As a side note, consider running the Best Practices Analyzer for SQL Server 2005 once you have the new server running. As of this writing, though, the BPA for 2005 is still only in a Community Technology Preview (read: beta) form, so any advice you get from it should be considered tentative until the finished version is out.

5. **Take the opportunity to re-optimize your SQL Server setup.** If you're migrating to a whole new edition of SQL Server and you're planning for some downtime anyway, don't just move everything as-is. Instead, see if you can take advantage of the moment to implement.

6. **optimizations that might help.** For example, if you've commissioned a new machine for SQL Server 2005 with more physical spindles than your previous machine, work out a new strategy for redistributing data tables and logs across those spindles. In addition, think about how new hardware (more and faster processors, etc.) will affect things like the MAXDOP setting. You might not have had to think about such things before because they simply weren't an issue, but what about now?

7. **Test and test again.** It's too easy to quit early when you're trying to verify that your newly-migrated database works. Don't take any behavior for granted -- make sure everything works as intended in as many different scenarios as possible. If you have an automated testing suite that you used with SQL Server 2000, dig it out again. This is especially important if you're migrating away from features in SQL Server 2000 that Microsoft doesn't support anymore because you want to find out if you transitioned away from them successfully.