

SQL Server 2005:

Preparing for a Smooth Upgrade

To meet a new generation of data-management needs, Microsoft® SQL Server™ 2005 has been reworked extensively to enhance performance and application programmability. Originally published by *SQL Server Magazine* as part of its “SQL Server 2005 Upgrade Handbook,” this article explores how administrators can help ensure a successful transition to SQL Server 2005 by planning, testing, and using the Upgrade Advisor.

BY ERIK VEERMAN

Related Categories:

Data Transformation Services (DTS)

Database administration

Database

Microsoft SQL Server 2005

SQL Server 2005 Integration Services (SSIS)

Visit www.dell.com/powersolutions for the complete category index.

Each Microsoft SQL Server 2005 component is designed to have a unique architecture and life cycle—the two primary areas that can affect an upgrade path. Some SQL Server 2005 components build on a solid foundation to augment, optimize, and help stabilize existing functionality. Microsoft has performed an extensive reworking of other SQL Server features to enhance performance and application programmability. SQL Server 2005 also incorporates completely overhauled components and additions designed to meet a new generation of data-management needs.

Preparing for a SQL Server 2005 upgrade involves understanding some basic principles that enable administrators to make appropriate decisions and help ensure success. As with any upgrade, the keys to success are appropriate planning and testing for the needs of the specific environment. This article explores the overall

upgrade path for SQL Server 2005 components and how the SQL Server 2005 Upgrade Advisor tool can help identify areas that require special attention. Specific upgrade considerations for certain SQL Server 2005 components—the database engine, Integration Services, Analysis Services, and Reporting Services—are also examined.

Upgrade mechanism

For all components, SQL Server 2005 provides an upgrade from SQL Server 2000 or SQL Server 7.0. Note that Microsoft distinguishes between a SQL Server 2005 upgrade and a migration.

An upgrade is an automated process in which the upgrade tool, called Setup, moves an old instance of SQL Server to a new instance while maintaining the data and metadata of the old instance. At the end of

SQL Server 2005	SQL Server 2000 or SQL Server 7.0
Database engine	<i>Upgrade tool:</i> Setup <i>Migration method:</i> Administrators perform side-by-side installation and then database backup/restore or detach/attach
Analysis Services	<i>Upgrade tool:</i> Setup <i>Migration tool:</i> Migration Wizard <i>Migration method:</i> Migration Wizard migrates objects, but optimization and client-access upgrades are required
Integration Services	<i>Upgrade tool:</i> None <i>Migration tool:</i> DTS Migration Wizard <i>Migration method:</i> DTS Migration Wizard converts 50 to 70 percent of the tasks, but some manual migration is required; runtime DTS DLLs are available in SSIS; package re-architecture is recommended
Reporting Services	<i>Upgrade tool:</i> Setup <i>Migration method:</i> Administrators perform side-by-side installation, and reports are deployed on the new instance
Notification Services	<i>Upgrade tool:</i> None <i>Migration method:</i> Upgrade of Notification Services instances occurs during installation

Figure 1. Upgrade path for Microsoft SQL Server 2005 components

the upgrade, the old instance is no longer available and the new instance has the same name as the old instance. Alternatively, migration is a manual process in which the database administrator installs a new instance of SQL Server and copies the metadata and data from an old instance of SQL Server to the new instance. Migration provides access to two instances of the system, letting administrators verify and compare the two systems. During migration, both the old and new systems remain online until migration to the new instance is complete. At the end of the migration, all applications are directed to access the new instance and the old instance is manually removed.

Although the SQL Server 2005 database engine introduces many features, administrators can easily upgrade databases on SQL Server 2000 or SQL Server 7.0 to SQL Server 2005 by using the Setup wizard or by performing a database restore or attach/reattach. Moving from Data Transformation Services (DTS) to SQL Server 2005 Integration Services (SSIS), however, requires a migration assisted by an out-of-the-box migration tool to help move data processing to the SQL Server 2005 architecture. Figure 1 summarizes the upgrade path for each SQL Server component.

Using the compiled knowledge from its product team, internal lab testing, and extensive SQL Server 2005 early-adopter

experience, Microsoft has developed an essential tool for upgrade preparation called the Upgrade Advisor. Figure 2 shows the Welcome screen for the Upgrade Advisor, which analyzes the configuration of the existing database server, services, and applications and provides reports that identify changes within the SQL Server product that can affect the upgrade. These changes include security enhancements, closer adherence to the SQL standard compared to previous SQL Server versions, and architectural changes. The Upgrade Advisor also provides links to documentation that describe these changes and necessary steps to complete the upgrade process. The Upgrade Advisor can help administrators manage the changes between releases, improve upgrade planning, and minimize any problems after the upgrade has completed. Whether enterprises are running Analysis Services for business intelligence, DTS for data processing, Notification Services for alerting, Reporting Services for enterprise reporting, or a combination of components, the Upgrade Advisor can help.

The Upgrade Advisor, built on a rules-based engine, is easy to install and run—even on remote servers. When administrators execute the tool, a simple wizard prompts them to select components on a local or remote server, as shown in Figure 3. Based on the selection, the wizard prompts administrators to identify details about each component. For the database engine, they can pick all the databases on the server or select them separately. The Upgrade Advisor analyzes all stored procedures and embedded Transact-SQL (T-SQL) programs. Furthermore, administrators can point to a SQL trace file that can analyze the T-SQL program running against the databases (an important

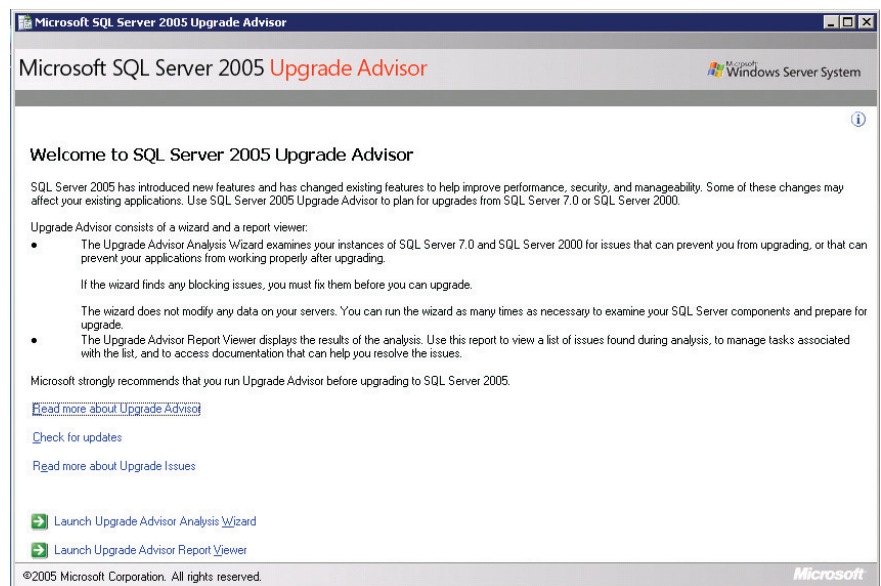


Figure 2. Microsoft SQL Server 2005 Upgrade Advisor Welcome screen

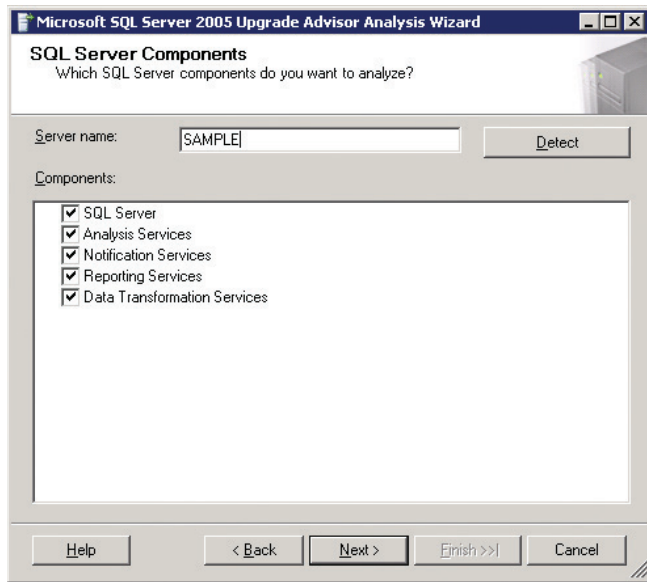


Figure 3. Selecting components for the SQL Server 2005 Upgrade Advisor to analyze

feature when running applications with embedded SQL logic). Administrators can analyze DTS packages that might be stored in files or embedded in the SQL instance they select; they can also select the Notification Services instance at this time.

After the tool completes its analysis, administrators can view a list of issues in the Upgrade Advisor Report Viewer. The Report Viewer provides a summary of issues, noting whether the corrections should be implemented before or after the upgrade. The Report Viewer lets administrators view the details of any server modifications they need to make, which objects (such as scripts or stored procedures) need to be modified, and details about when to make the changes. The Report Viewer also helps administrators manage the modification tasks—letting them check off completed tasks, sort tasks, and create Microsoft Excel spreadsheets of the report details to distribute among members of the project team.

In addition, the Upgrade Advisor lets administrators drill down into the report details, opening a Microsoft Help file that explains how to address specific issues and workarounds. After administrators view the details of a specific issue, they can browse to other rules included in the Help file and see additional areas that the tool evaluates during its analysis. Microsoft includes the Upgrade Advisor with the SQL Server 2005 server installation CD.

Note: Administrators should review the included readme file before installing the Upgrade Advisor; this file contains crucial information about the required prerequisite software and a description of the tool’s included rules, known issues, and so forth.

Upgrade process

The Microsoft SQL Server upgrade process can be broken down into four phases: planning and research, testing and process validation, production upgrade, and post-upgrade. This section examines the first three phases. For post-upgrade considerations, see the supplemental online section of this article at www.dell.com/powersolutions.

Planning and research

Developers, database administrators, and application architects should have sufficient resources to start the educational and review process. Their training, experience, and research can drive much of the planning process. Because they intimately know the applications’ profiles, they can provide valuable insight into the upgrade details.

The planning phase should move from identifying the databases targeted for the upgrade to determining the changes and processes the upgrade will require. The Upgrade Advisor can help the team determine where to focus its efforts and what to expect. A major decision in this preliminary phase is to decide whether to perform an in-place upgrade or a side-by-side migration. Administrators should base this decision on a combination of factors, including the platform upgrade path available, enhancements to implement during the upgrade, the application architecture, and hardware requirements.

Generally, enterprises should conduct the following planning activities:

- **Learn about the SQL Server 2005 upgrade tools:** Administrators should understand the platform’s highlights, examine the functionality, and test the upgrade and migration tools.
- **Assess the application features:** Administrators should evaluate and determine which applications, servers, and databases can benefit most from the upgrade.
- **Select the upgrade path:** Administrators can use the Upgrade Advisor to help determine which upgrade path, in-place upgrade or side-by-side migration, will work best for the environment.
- **Identify the prerequisites for the upgrade process:** Administrators should work with the upgrade team to research compatibility and functionality changes that can help ensure a successful upgrade and can take advantage of the release’s enhanced features. The Upgrade Advisor can provide valuable help.
- **Set specific planning and research milestones:** Administrators should determine the upgrade path and steps, set up an initial test plan, and implement a risk mitigation and recovery plan.

Testing and process validation

Nothing can replace testing. Even if administrators plan to upgrade only the SQL Server database engine without changing the application, testing can help identify any backward-compatibility problems and behavioral changes from previous SQL Server releases that the Upgrade Advisor did not detect. Furthermore, testing can help validate data and organize the upgrade process. This phase entails establishing a test environment and composing validation scripts and application functions to confirm a successful upgrade.

The final plan should include a backup of the SQL Server 2000 or SQL Server 7.0 databases and a tested recovery strategy. Also, administrators should identify all application references (such as connection strings, package references, and reports) to the upgraded SQL Server components. For this task, an in-place upgrade offers advantages over a side-by-side migration: When administrators upgrade an earlier SQL Server release in-place through the installation upgrade process, all existing application connections remain the same because the server and the server instance do not change.

Enterprises should conduct the following tasks in the testing and validation phase:

- **Prepare the test environment:** Side-by-side migrations require a separate test SQL Server 2005 installation. In-place upgrades require a test machine running SQL Server 2000 or SQL Server 7.0 and target database copies; hardware comparable to the production setup can allow for production volume testing.
- **Set a pre-upgrade baseline:** This baseline can help administrators evaluate the system post-upgrade and determine any behavioral changes, letting them simulate a typical workload after the upgrade. The baseline can also help administrators confirm functionality and document performance improvements or changes. To set up the baseline, administrators can use familiar tools such as SQL Server Profiler, application load testing tools, Performance Monitor counters, and Showplan statistics.
- **Develop a test plan:** Administrators should set up a generalized testing script or test procedures for the following areas: data validation, data processing, stress and workload, client/server performance, and application functionality
- **Develop a recovery plan:** Administrators should develop upgrade rollback procedures in case of an upgrade interruption. The recovery plan should include running a Database Console Command (DBCC) consistency check on the pre-upgrade databases before backup as well as performing a full restore of the database to validate the backup reliability. After the upgrade, administrators should perform a consistency check and a backup with validation. They also should make sure to test the rollback procedures.

- **Create application-modification procedures:** The test environment should include the full application tier so administrators can confirm that application changes work as expected. These application-modification procedures should include a catalog of affected users. Such procedures also allow for complete documentation of application changes so that they can be applied successfully during the production cutover.
- **Perform an upgrade test run:** A final test run of the upgrade can confirm that the process and procedures work as expected. Administrators can use the Upgrade Advisor after applying the pre-upgrade changes to validate that they have addressed all the problem areas the tool identified.

Production upgrade

The SQL Server 2005 Upgrade Advisor and Setup wizard are designed to help administrators proceed confidently through the planning and testing steps, positioning them for a successful production upgrade. Administrators can use some of the testing steps developed for pre-upgrade use (such as record counts and validation scripts) in validating the upgrade upon completion. Generally, enterprises should perform the following steps for the production upgrade, depending on the SQL Server components being upgraded:

1. **Back up the systems (applications and databases).** Perform a consistency check if applicable, back up the database and related systems, and then validate the backup.
2. **Perform pre-upgrade tasks.** Notify users and then disable the user interface components, pausing all data processing, data entry, and data changes. Make the necessary pre-upgrade changes identified during the testing phase. Re-execute the Upgrade Advisor to validate the pre-upgrade state, and perform an optional secondary backup of the systems before the upgrade.
3. **Perform primary SQL Server back-end platform upgrade tasks.** Run SQL Server 2005 for a side-by-side migration. Install the Microsoft .NET Framework and SQL Native Client. In the Setup wizard, specify the same instance as the legacy installation. Then, specify the same components as the legacy instance (for example, Database Services, Analysis Services, and Reporting Services). Once the setup is complete, perform the tasks required for special upgrade considerations (such as repopulation of full-text indexes, special handling of clusters, or log shipping). Next, make any post-upgrade platform changes, such as scripts or tasks required to support the back-end functionality on the new SQL Server 2005 platform. Finally, run platform data and functionality validation testing scripts to confirm the success of the SQL Server 2005 upgrade.

- 4. Make primary application changes.** Make application functionality changes to support the new back-end structures, and make any required database reference changes in application connection strings and other connection references. Test application functionality, including data processing, front-end and report usage, and other application components based on the test procedures created in the planning phase.
- 5. Perform post-upgrade steps.** For the database engine, the upgrade automatically sets the compatibility mode to 8.0; however, administrators may wish to switch to compatibility mode 9.0 to take advantage of the features introduced in SQL Server 2005. For side-by-side migrations, stop the former platform services (or set the database to read-only) to prevent unknown data changes. For the relational data, run DBCC consistency checks to validate the data. Back up SQL Server 2005 structures and data with backup validation, and back up application systems and files. Then, re-enable processing and the application user interface, notifying users that the upgrade is complete.

Upgrade considerations for specific SQL Server 2005 components

Given the breadth of the Microsoft SQL Server 2005 platform, upgrade processes vary for different components. This section examines the upgrade considerations for major SQL Server 2005 components.¹

Upgrading to the SQL Server 2005 database engine

The database engine is the easiest to upgrade of all SQL Server components, and upgrading it can provide an immediate return on investment in the areas of management, performance, and high availability. The two main options for the database engine upgrade are side-by-side migration (in which the SQL Server 2005 engine is installed as a secondary instance on the same server as the SQL Server 2000 or SQL Server 7.0 engine or on a completely separate server) and an in-place upgrade (in which an instance of SQL Server 2000 or SQL Server 7.0 is upgraded through the installation process and databases and other objects are upgraded “in place”).

With a side-by-side migration, the most common upgrade path is a simple database detach and re-attach on the SQL Server 2005 instance or a database backup and restore from the older version to the new version. If administrators retain an up-to-date version of the metadata scripts, they also can create the objects on the SQL Server 2005 server and use the bcp utility to export and import the data. The other option is an in-place upgrade, in

which administrators upgrade and adapt the databases, settings, and extended features to the SQL Server 2005 engine during the installation process; when running the setup process on a server that has a SQL Server 2000 or SQL Server 7.0 instance, administrators should see an option to upgrade the selected instance to SQL Server 2005.

Note that for the database engine upgrade, all existing Microsoft Data Access Components (MDAC) and ADO.NET applications should continue to function as when they were running on SQL Server 2000 or SQL Server 7.0. In fact, SQL Server 2005 does not include an updated release of MDAC. However, SQL Server 2005 introduces the SQL Native Client, which combines an updated SQL Open Database Connectivity (ODBC) driver and SQL OLE database (OLEDB) provider with network libraries in a single dynamic-link library (DLL). The SQL Native Client lets administrators leverage the SQL Server 2005 client-access features, such as Multiple Active Result Sets (MARS), the XML data type, and user-defined types (UDTs). SQL Server 2005 provides tight integration with the Microsoft .NET Framework 2.0, which includes the latest ADO.NET version.

The in-place server upgrade typically is easier to perform than the side-by-side migration. Although this approach requires a more thorough fallback plan and testing, it also provides seamless connectivity. By performing an in-place upgrade, logins and users remain in sync, database connections remain the same for applications, and SQL Agent jobs and other functionality are concurrently upgraded during the installation. Note that several features, such as log shipping, replication, and cluster environments, have special upgrade considerations.

For the database engine, the upgrade sets the compatibility mode to 8.0. Keeping this setting at 8.0 may be beneficial for certain circumstances, such as for T-SQL references that are no longer supported in SQL Server 2005. The analysis phase of the upgrade process should uncover situations in which using a compatibility setting lower than 8.0 may be preferable. However, best practices recommend fixing any syntax that requires a compatibility level lower than 9.0 (SQL Server 2005) during the upgrade process. By reworking the syntax, developers can have immediate access to the programming enhancements and features in the SQL Server 2005 release. To isolate these type of issues and other syntax that can cause upgrade trouble, administrators can script out the objects and procedures from the previous platform version and attempt to run the scripts within SQL Server 2005. A simple attach or restore might suppress these issues. Also, some SQL logic can be embedded in the application. For data validation, administrators can run the DBCC checkdb statement on the attached or restored database to confirm the integrity of the migrated data.

¹ For more information about Microsoft SQL Server 2005 upgrade considerations, see the white papers, Webcasts, and other resources listed at www.microsoft.com/technet/prodtechnol/sql/2005/sqlupgrd.mspx#ECAA.

Note: Microsoft recommends using Information_Schema views to obtain various metadata instead of querying the system tables directly because Microsoft cannot guarantee that the underlying object structure will persist in new platforms. With the release of SQL Server 2005, Microsoft has changed the SQL Server underlying object structure. Also, SQL Server 2005 catalog views and Dynamic Management Views (DMVs) have restricted permissions. PUBLIC users no longer have permissions to view catalog views, and users with GUEST/PUBLIC permissions cannot select from DMVs.

Migrating to SQL Server 2005 Integration Services

Microsoft did not use the name of the SQL Server 2005 Integration Services predecessor, Data Transformation Services, for its SQL Server 2005 extraction, transformation, and loading (ETL) component because SSIS was a complete code rewrite—Microsoft did not use the DTS code to create this component. With industry demands for fast performance and hardware consolidation to handle ever-increasing data complexity and volume, DTS was not positioned as a long-term solution. Although DTS and SSIS are both ETL tools, their architectures diverge greatly. Because of this, moving from DTS to SSIS requires a migration, which involves redesign and solution changes to leverage the SSIS features.

The migration from DTS to SSIS uses wizard-driven output along with manual redesign. Some DTS tasks have a straightforward upgrade path to SSIS and are accommodated by the DTS Migration Wizard. Administrators may be able to use this wizard to upgrade other tasks depending on their use and design, but some tasks may be more difficult to upgrade or not upgradeable. The supplemental online section of this article, available at www.dell.com/powersolutions, describes some issues that administrators may encounter when upgrading DTS packages.

Administrators can incrementally migrate packages to SSIS. When installing SQL Server 2005, they have the option to install the runtime files required for DTS packages to execute on SQL Server 2005—without SQL Server 2000 needing to be installed. This makes the side-by-side migration appealing, especially in an environment where the DTS packages contain many tasks that require manual migration. SSIS also contains an Execute DTS package object when the runtime files or SQL Server 2000 has been installed on the SSIS server. A side-by-side implementation of SQL Server 2005 SSIS and SQL Server 2000 DTS can provide flexibility as administrators approach package migration.

Upgrading Analysis Services

Dimensions, partitions, storage modes, aggregates, and measures—the strengths of Analysis Services 2000—have been preserved in the Analysis Services 2005 release. However, SQL Server 2005 also brings many notable enhancements. The Unified

Dimensional Model (UDM), for example, extends beyond traditional online analytical processing (OLAP) sources to allow expanded relational and aggregate data in a unified view. Dimensions are another area with valuable changes from previous SQL Server versions. A shift from a hierarchy-based model to an attribute-based model, with related optimizations on the storage and aggregation side, allows Analysis Services 2005 to scale for enterprise performance and volume.

From an upgrade perspective, Microsoft provides a direct in-place upgrade from Analysis Services 2000 to Analysis Services 2005—preserving cubes, partitions, dimension hierarchies, measures, calculations, and sets. Because Analysis Services objects are built on top of a Data Source View (DSV) referencing database engines, best practices recommend creating the DSV on the base tables that the Analysis Services 2000 objects are built on rather than on views referencing underlying tables. The Migration Wizard generates DSVs that are complete with relationships and attributes from source tables. This can allow developers to add attributes to the cube even though they were not present in Analysis Services 2000.

Note that the Migration Wizard does not optimize the Analysis Services objects; it simply moves the objects in place to the new Analysis Services server. The goal of the wizard is to migrate the cube structures and architecture objects so that client applications relying on the Analysis Services 2000 structures do not fail after administrators have migrated the cube to Analysis Services 2005. Thus, the migrated cube design may not take advantage of SQL Server 2005 enhancements. However, the cubes should have the immediate performance and scalability benefits of the Analysis Services 2005 architecture. When the Migration Wizard finishes its processes, administrators can then reprocess the cube and test the data and reports.

For Analysis Services 2005, the major upgrade considerations revolve around the client-access methods and structure impact to reports. Analysis Services 2005 takes advantage of the Web service protocol for OLAP—XML for Analysis (XML/A)—that Microsoft helped write. (Support for XML/A was available for Analysis Services 2000 as a Web release, letting an Analysis Services 2000 server listen and respond to XML/A requests.) With native support for XML/A in Analysis Services 2005, administrators should update existing client components of OLEDB for OLAP (Pivot Table Services, or PTS) to access SQL Server 2005. That means users require the latest version of PTS that is included with SQL Server 2005. The new driver should be installed side-by-side with the earlier PTS version, letting users access both SQL Server 2005 and SQL Server 2000 Analysis Services.


The second client-access consideration is the OLAP structure and related Multidimensional Expression (MDX) compatibility after the upgrade. MDX is not gracious to members and structures that

have changed. Although the Upgrade Wizard sufficiently recreates the OLAP structure, with the dimension architecture change from hierarchy based to attribute based, administrators may find small structural and data anomalies that appear after the upgrade. Therefore, report and data testing are more critical here than on the database engine side. Administrators may need to recreate reports and underlying MDX for the structures in Analysis Services 2005.

Upgrading Reporting Services

Because Microsoft initially released Reporting Services 2000 in early 2004, the Reporting Services 2005 platform does not introduce major architectural changes, but it does offer features such as multi-select parameters, built-in MDX support, and dynamic report generation. Microsoft provides a direct, in-place upgrade path for moving from Reporting Services 2000 to Reporting Services 2005. Furthermore, Reporting Services 2005 runs Report Definition Language (RDL) report definitions created in Reporting Services 2000 without requiring administrators to upgrade the definitions. However, when developers open a report in Business Intelligence (BI) Development Studio, they are prompted to convert the RDL definitions to the Reporting Services 2005 standards.

A tool for successful upgrades

Managing the upgrade to Microsoft SQL Server 2005 requires significant planning and testing. With appropriate forethought and preparation—and use of the SQL Server 2005 Upgrade Advisor and Setup wizard—administrators can avoid problems and identify the areas where they need to concentrate their efforts. After performing a smooth upgrade, administrators can be ready to fully leverage the power and functionality that SQL Server 2005 is designed to provide. 

Erik Veerman is an associate mentor for Solid Quality Learning and has designed several SQL Server–based business intelligence solutions across a broad business spectrum. Erik—an expert in OLAP design, ETL processing, and dimensional modeling—is a frequent presenter for his local Professional Association for SQL Server (PASS) chapter and speaks at the national PASS and *SQL Server Magazine* Connections conferences.

Edited with permission from SQL Server Magazine. Copyright © 2005 Penton Media, Inc. All rights reserved.