Chapter 2

New Features in SQL Server 2005

In This Chapter

- Checking out SQL Server's improved security features
- Finding out about other SQL Server 2005 enhancements

SQL Server 2005 is the most exciting release of SQL Server for years. It may be the only release in the last five years, but it's genuinely an exciting release full of new and useful features. In addition to new features, SQL Server 2005 also includes many features that are big improvements on their counterparts in SQL Server 2000.

In this chapter, I briefly describe many of SQL Server 2005's new and improved features.

To get a good feel for the range of new and improved features in SQL Server 2005, be sure to read all sections of the chapter.

To describe all these features in detail and explain how to use them would need a book maybe ten times as long as this one. I had to make choices about which tools or features to cover in detail later in the book. I often assume that you have some familiarity with SQL Server 2000, but you should be able to follow along even if you are new to SQL Server 2005. For many topics I tell you where I describe the functionality in more detail.

Because one of the most important aspects of any database is security (a topic I explore further in Part IV), I start there.

Security Enhancements

Security is a major focus for new features in SQL Server 2005. In part, this focus reflects a response to issues such as the Slammer worm that hit SQL Server 2000. In part, it reflects a world where more business data is potentially exposed on the Internet. SQL Server has to give you the tools to keep your data safe in order to allow the right people to access data that you want them to access and to stop other people from accessing data that you don't want them to access.

I describe security in SQL Server 2005 in more detail in Chapter 11.

System catalog security

The system catalog in SQL Server 2005 consists of views of the underlying system data structures. Users do not see any underlying tables, so unskilled or malicious users can't change or otherwise corrupt them. This stops you or anyone else from damaging the core structures on which your SQL Server installation depends.

Password policy enforcement

When installed on Windows 2003 Server, you can apply to SQL Server 2005 any Windows password policy that you have in effect. You can enforce policies for password expiration and strength on SQL Server 2005 in exactly the same way as for Windows logins. Windows 2000 Server does not support this.

You can turn off (or on, in some cases) password policy enforcement for individual logins. For example, you can turn off password policy enforcement when you're using an application with built-in authentication information that you can't change.

I show you how to alter password enforcement in Chapter 11.

Schema and user separation

SQL Server 2000 had no concept of a schema: A user owned a database object. So if a user `User1` created an object called `myTable`, then the object's qualified name was `User1.myTable`. If `User1` is deleted — for example, when the individual leaves the company — you needed to change the name of the object, which caused problems for applications that depended on the name of the object for data access. In SQL Server 2005, a user can create a schema, which in turn contains database objects, which has a different name from the user. `User1` can create a schema called `HR.` and create an object called `myTable`. You refer to that object as `HR.myTable`. So if `User1` leaves the company, you can leave the schema name unchanged, which means you can leave your application code unchanged because the object is still called `HR.myTable`.

Automated certificate creation for SSL

In SQL Server 2000, when using *Secure Sockets Layer* (SSL) to log in to a SQL Server instance, you had to manually create a certificate to underpin the use of SSL. SQL Server 2005 creates a certificate automatically. That allows you to use SSL without manually creating a certificate.

Transact-SQL Enhancements

SQL Server 2005 has added several new features to Transact-SQL.

REMEMBER

Transact-SQL is the version of the Structured Query Language (SQL) used by SQL Server 2005. Transact-SQL is often abbreviated to *T-SQL*. T-SQL has many features, which are not included in ANSI SQL.

Improved XML support

SQL Server 2000 allowed you to retrieve relational data as XML with the `FOR XML` clause or store XML as relational data in SQL Server, using the `OPEN XML` clause. SQL Server 2005 has a new `xml` datatype that allows you to write code to retrieve XML data as XML, avoiding the transformation from XML to relational data that occurred when using `OPEN XML`. You can also use a schema document expressed in the W3C XML Schema Definition language (sometimes called XSD schema) to specify allowed structures in the XML.

Note: Strictly speaking, the `xml` datatype stores data in a proprietary binary format. For practical purposes, you can retrieve and manipulate the data as XML.

The `xml` datatype supports several keywords in T-SQL. I show you how to use these keywords in Chapter 7.

Error handling

SQL Server 2005 allows you to use `TRY ... CATCH` blocks in your T-SQL code. So, if your code causes an error, the code in the `CATCH` block allows you to specify what to do when an error occurs. I show you how to use `TRY ... CATCH` blocks in Chapter 10.

Transact-SQL templates

In SQL Server Management Studio, you can find many templates to help you carry out common tasks with Transact-SQL. To view the range of T-SQL templates, choose View➪Template Explorer in SQL Server Management Studio. The Template Explorer displays. View the nodes to see the range of templates available. Check out Chapter 5 for more on templates.

Other Developer-Orientated Enhancements

SQL Server 2005 has several new developer-orientated enhancements. I describe those in the following sections.

Support for the Common Language Runtime

The Common Language Runtime (CLR), which is used by .NET code, is embedded in the SQL Server 2005 database engine. You can write stored procedures, triggers, functions, aggregates, and user-defined datatypes by using languages such as Visual Basic .NET or C#. Stored procedures written in a .NET language are a good replacement for SQL Server 2000 extended stored procedures, because you can specify a security level for the .NET code.

You find three security levels for .NET code:

- **Safe:** This level allows no access outside SQL Server. For example, your code cannot access the file system, registry, environment variables, or the network. This security level is the most secure.
- **External Access:** This security level allows limited external access by your code. Specifically, you can access the registry, the file system, environment variables, and the network.
- **UnSafe:** You can access any desired functionality outside SQL Server 2005 with the UnSafe security level. You should use the UnSafe security level only if you are certain that the code is well written and you trust the author of the code.

New datatypes

SQL Server 2005 supports several new datatypes:

- **`varchar(max)`:** This allows you to use sequences of characters greater than 8000 bytes (8000 characters). The maximum size is 2GB.
- **`nvarchar(max)`:** This allows you to use sequences of Unicode characters greater than 8000 bytes (4000 characters). The maximum size is 2GB.
- **`varbinary(max)`:** This allows you to use binary data greater than 8000 bytes.

Each of the preceding datatypes can be up to 2GB in size. This allows major size increases in size compared to `varchar(8000)` and `nvarchar(8000)`, which were each limited to 8K.

SQL Management Objects (SMO)

SQL Management Objects (SMO) replaces Distributed Management Objects (DMO), which were used in SQL Server 2000. SQL DMO applications run on SQL Server 2005 but no updating of DMO objects took place for SQL Server 2005. Developers use SMO. However, applications created with SMO often provide custom management tools for administrators.

SMO is faster than SQL Server 2000 DMO in many settings because each object is only partially instantiated. For example, if you want to enumerate what might be thousands of databases on a powerful server, you don't need fully instantiated objects to populate a tree view. You need only the object's name. Having partially instantiated objects saves a lot of time for commonly used, simple tasks because you probably need a fully instantiated object for only a small number of the total number of objects.

Scripting actions

If you have used Microsoft programs such as Access and Excel you'll know that you can create macros to allow you to automate certain tasks. SQL Server 2005 now has a feature that automatically creates Transact-SQL scripts from actions you take using the graphical user interface in SQL Server Management Studio. You can use these scripts exactly as SQL Server Management Studio creates them or you can modify them in ways to exactly suit your intentions.

HTTP endpoints

HTTP access to SQL Server 2005 is a new feature that allows programmers to access SQL Server without depending on an IIS server running on the same machine. SQL Server can coexist with IIS but unlike with SQL Server 2000, IIS is no longer required for SQL Server 2005. HTTP endpoints allow developers to use XML Web services with SQL Server 2005. The HTTP endpoint can execute T-SQL batch statements or stored procedures.

For security reasons, HTTP endpoints are disabled by default. To use HTTP endpoints you need to specify which users, stored procedures, and databases are enabled to support it.

Manageability Enhancements

The management tools in SQL Server 2005 have changed greatly from SQL Server 2000. The main change is the arrival of SQL Server Management Studio, which I describe in more detail in Chapter 3.

New management tools

SQL Server 2005 has new management tools. SQL Server Management Studio replaces Enterprise Manager and Query Analyzer that you may be familiar with from SQL Server 2000. SQL Server Management Studio also allows you to manage Analysis Services instances and, therefore, also replaces Analysis Manager.

SQL Server Management Studio allows you to manage multiple SQL Server instances more easily. From one interface you can manage multiple instances of the SQL Server database engine, Analysis Services, Integration Services, and Reporting Services. I describe SQL Server Management Studio in detail in Chapter 3.

SQL Server Configuration Manager is a new tool that allows you to control services associated with SQL Server 2005. SQL Server Configuration Manager replaces Service Manager and the server and client networking tools. You can use SQL Server Configuration Manager to control the following:

- SQL Server
- SQL Agent
- SQL Server Analysis Services

- DTS Server (for SQL Server Integration Services)
- Full-text Search
- SQL Browser

Profiler

Profiler has many features that allow you to analyze performance problems in SQL Server 2005. For example, Profiler opens trace files that you store in the file system, which allows you to replay and analyze interesting SQL Server processes. Profiler can display a graphical representation of a trace so you can easily see what is happening.

Profiler can import data recorded by using the Windows Performance Monitor. You can display the data graphically, letting you see performance over a selected period of time. From the graph, you can enter the trace at the point where a problem lies — for example, where CPU usage spikes. You can then closely examine what is causing a performance problem.

I don't cover Profiler in detail in this book.

SQL Server Agent

The capabilities of SQL Server Agent, the component that supports scheduled jobs, have been enhanced. For example, the number of concurrent jobs that SQL Server Agent can run has increased. SQL Server 2000 used SQL Agent only in relation to jobs for the database engine. SQL Server 2005 also uses SQL Server Agent to run jobs for Analysis Services and Integration Services. I discuss SQL Server Agent jobs further in Chapter 16.

SQL Server Agent uses Windows Management Instrumentation (WMI). WMI support allows you to write code to avoid running a job, such as when the disk space is insufficient to let the job run successfully.

Dynamic configuration

In SQL Server 2005, you can make many configuration changes without having to restart SQL Server (if you're running on the Windows Server 2003 operating system). This is a big improvement over SQL Server 2000 when you often had to restart SQL Server after making configuration changes. In SQL Server 2005, you can change CPU affinity or I/O affinity if you need to. If you have the necessary hardware you can hot-add extra memory to your server.

Full-text search

You can back up and restore Full-text Search catalogs (the databases where the metadata is stored) in the same way as you back up and restore any other SQL Server 2005 databases.

SQL Server Service Broker

SQL Server Service Broker allows you to create asynchronous message-based applications. The asynchronous nature of Server Broker messages means that processing tasks needn't be done all at the same time, which can slow the server down. Instead, messages are queued for processing when the server load is lighter, improving overall performance.

Turn to Chapter 19 for more info on Service Broker.

Dedicated Administrator connection

This is not a connection only for dedicated administrators but a connection that only administrators can use. The connection is used when a runaway process is on the server and you need access to the database engine to kill the process. Even if the runaway process is using close to 100 percent of the CPU cycles, the Dedicated Administrator connection allows administrators to get a share of CPU cycles and so kill the runaway process. This capability allows you to kill a runaway process without having to restart the server.

SQLCMD

The SQLCMD command-line utility is the recommended command-line tool in SQL Server 2005. You can write SQLCMD commands individually or use SQLCMD to execute T-SQL scripts.

The SQLCMD utility allows you to use command-line parameters to replace variables in a T-SQL script. For example, suppose you had a simple backup script

```
BACKUP DATABASE $(db) TO DISK = "$(path)\$(db).bak"
```

called `backup.sql`. It backs up a specified database to a specific location. You can run that script from the command line, replacing the `$db` and `$path` variables with the relevant database name and file path. For example, to back up `AdventureWorks` into the C:\Backups folder, you write:

```
sqlcmd -E -i Backup.sql -v db="AdventureWorks"
          path="C:\Backups"
```

Easier updates

System objects are in the `resource` database in SQL Server 2005. When a service pack or other update is applied to a SQL Server instance, replacing the `resource` database updates all the system objects. This gives a more easily manageable upgrade path.

Replication

The setup and administration of replication has been improved in SQL Server 2005. This is carried out from SQL Server Management Studio. There is a new system health monitor to allow you to check on replication settings and performance. For example, it can tell you how long replicated data takes to reach subscribers.

WMI configuration

Windows Management Instrumentation (WMI) allows you to carry out a range of configuration tasks such as specifying client and network settings. You can also use WMI to determine whether you can safely and effectively carry out a task at a particular time. For example, you can test whether a target drive has enough disk space to carry out a backup.

Database Mail

Database Mail is a new feature. It replaces SQLMail, which was present in SQL Server 2000. Database Mail uses Simple Mail Transfer Protocol (SMTP). There is no longer any dependency on the Messaging Application Programming Interface (MAPI), and Outlook is no longer required. The removal of these dependencies avoids many of the availability problems that SQL Server 2000 users had with SQLMail. In addition, Database Mail is cluster aware, unlike SQLMail. Database Mail supports logging and auditing.

Availability Enhancements

SQL Server 2005 supports three broad types of enhanced availability:

- Concurrent data access
- Availability after server failure or other disaster
- Availability during database maintenance and repair

Concurrent data access

Concurrent data access is about getting to data when others are also accessing it:

- **Database snapshots:** Creates a logical copy of a database at a specific point in time. You can use snapshots as the basis for Reporting Services reports without putting a load on the live copy of the database.
- **Snapshot isolation:** This is a new transaction isolation level that improves availability for read applications, because writes do not block reads. Write applications are subject to mandatory conflict detection.
- **Online Indexing:** During the time that an index is being rebuilt, it remains possible to use the index for accessing data. This is likely to improve performance during that period.

Availability after server failure

SQL Server 2005 provides better availability in scenarios relating to server failure or some other catastrophic situation. The following features are additional to server failover clustering that was supported in SQL Server 2000 Enterprise edition:

- **Faster Recovery:** After a server failure, databases are made available to users more quickly during the recovery process, which improves availability.
- **Database Mirroring:** Using standard hardware, a mirror server always maintains an up-to-date copy of the database. On failure of the principal server, the mirror server is available within three seconds, making the server failure invisible to users.
- **Transparent Client Redirect:** Under the covers, Database Mirroring uses Transparent Client Redirect. The MDAC (Microsoft Data Access Components) layer notes the mirror server when connecting to a principal server. If the principal server fails, MDAC redirects the connection to the mirror server.

Availability during database maintenance

The backup and restore process must be reliable if you are to restore a database effectively, such as after a hardware failure. Here are the relevant features of SQL Server 2005:

- **Finer Grained Online Repairs:** You can restore data filegroup by filegroup with the database being available after the primary filegroup is restored.
- **Enhanced Backup Verification:** The verifying of backups is more complete than in previous versions of SQL Server. The `RESTORE VERIFY ONLY` syntax checks everything that can be checked, short of writing the backup to the server.
- **Backup Media Mirroring:** This allows you to make extra copies of backups to minimize the chance of a failed restore or to make an archival copy for offsite storage, for example.
- **Database Page Checksums:** Checksums are added to individual pages in the database to detect errors that otherwise might go undetected.
- **Backup Checksums:** Adds an additional error detection mechanism during backup. It is possible to proceed past errors and fix them later.
- **Backup Data and Logs:** The former problems with log backups at the same time as data backups have been remedied. It is now possible to carry out a data backup at the same time as the corresponding log backup.

Scalability Enhancements

One of the key aims of SQL Server 2005 is to improve the scalability of databases and the applications that depend on them.

Installing in a cluster

You can install SQL Server 2005 in a cluster of up to eight nodes with status reporting of install progress on each node. SQL Server 2005 setup has the ability to install Analysis Services in a cluster. New in SQL Server 2005 is the ability to carry out unattended installation to a cluster.

Partitioning data

Some database objects are very large. Partitioning splits large database objects into multiple more manageable pieces. Suppose you have a huge table that contains many months of data. You could improve performance by partitioning the table into several smaller partitions with each partition containing the data for a single month.

You can also partition indexes. If the partitions for data and for indexes are aligned, you can move partitions into or out of a table. This is useful if you have data for the last 12 months, partitioned by month. You can move the data for 12 months ago out and create a new partition for the current month.

Database Engine Tuning Advisor

The Database Engine Tuning Advisor (DTA) is the SQL Server 2005 replacement for the SQL Server 2000 Index Tuning Wizard. The Database Tuning Advisor can help you tune performance for the whole database, not just for indexes, as previously.

The DTA can handle partitions and can assist in tuning some operations that involve using multiple databases. It has a high availability recommendation mode that recommends creation of indexes only if those indexes can be built online, which improves availability of a database during creation of indexes. You can specify a maximum time for DTA to reach its recommendations to avoid very long running scenarios. You can also explore `what if` analyses in the Database Tuning Advisor to allow you to explore the effects of possible approaches.

Hot-add memory support

If the demands on your SQL Server increase markedly and you have the appropriate hardware and are running SQL Server on Windows Server 2003, you can take advantage of hot-add memory support. This allows you to improve performance under high load and also allows you to avoid downtime.

Replication

Replication performance has been improved in SQL Server 2005. The number of subscribers supported in replication is increased.

Business Intelligence Enhancements

Business Intelligence is a prominent strength of SQL Server 2005. Features have been added to Analysis Services, Integration Services has replaced Data

Transformation Services, and Reporting Services builds on the features of Reporting Services 2000 that was first released early in 2004.

Business Intelligence in SQL Server 2005 uses an Integrate, Analyze, and Report paradigm. You can use Integration Services to bring data together from various sources. Use Analysis Services to provide insight into the data. You can create reports to present the analyses to business users with Reporting Services.

Integration Services

SQL Server Integration Services replaces SQL Server 2000 Data Transformation Services. You can use Integration Services to import and restructure data. You can also load data into a data warehouse with Integration Services. I describe Integration Services in detail in Chapter 20.

Analysis Services

SQL Server 2005 extends Analysis Services functionality. You use Analysis Services to extract information that is meaningful in a business context. Analysis Services contains two major parts: OnLine Analytical Processing (OLAP) and Data Mining. I describe Analysis Services in more detail in Chapter 21.

Reporting Services

You can use SQL Server Reporting Services to create customized reports for end users. Reporting Services for SQL Server 2000 was introduced in January 2004, but was originally intended as a new feature in SQL Server 2005. If you are a developer, you create report projects in the Business Intelligence Development Studio. If you have the Enterprise Edition of SQL Server 2005, you can create a report model project in the Business Intelligence Development Studio, and your end users can use report model projects to create ad hoc reports with the new Report Builder design tool. I describe Reporting Services in more detail in Chapter 22.