Part 1

Introducing SQL Server

In this section:

- **Chapter 1: Introduction to SQL Server 2005**
- **Chapter 2: Overview of Database Concepts**
- **Chapter 3: Overview of SQL Server**
- **Chapter 4: Database Design and Normalization**

Chapter 1

Introduction to SQL Server 2005

Welcome to SQL Server 2005. In this book, we'll help you learn the basics of SQL Server and advance to more complex skills. You won't learn *everything* about Microsoft's flagship database here: It's a huge set of programs that can take years, if not decades, to learn fully. However, we'll show you how to get up and running quickly and how to handle typical everyday tasks of keeping your data safe, secure, and available to your users.

SQL Server 2005 isn't just an upgrade of SQL Server. Originally planned for 2003, the two extra years of work on Yukon (Microsoft's development code name for SQL Server 2005) have resulted in a product that is so different and so much better than its predecessor as to make you think the only things the two version have in common are similar names and use of SQL.

This isn't just hype or a marketing device. Microsoft has overhauled nearly every aspect of SQL Server. What we have today is a product that is more fully integrated, flexible, and extensible than any relational database product.

Before we dig deep into the details of SQL Server, we want to introduce you to the product. You may be a budding database administrator (DBA), anxious to manage a database for others to use; you may be a developer, ready to write code that will extract information from a server someone else is maintaining; or you may be a regular user who just needs to see some data and doesn't have time to wait for the IT department to build an application.

Whoever you are, *Mastering SQL Server 2005* has something for you. In this chapter, we'll talk briefly about the ways you'll probably work with SQL Server 2005, whether you're a DBA, a developer, or a user. Users in particular should review these pages even if you don't expect to ever write a line of code or manage a database. By knowing what *can* be done with SQL Server 2005, you'll be in a much better position to discuss with your IT people the sort of solutions and assistance you need. You may even discover that SQL Server 2005 can help you solve that problem you've been having with your data.

In this introductory chapter, we'll also look briefly at most of the new enhancements in SQL Server 2005. Of course, we can't highlight all the features or all the changes in SQL Server 2005, but we can show you enough to make you as excited about this database management system as we are.

The Editions of SQL Server 2005

Because SQL Server 2005 is used by a vast audience of different people—businesses, school, government agencies, and so on, all of whom have different needs as well diverse requirements—it comes in different editions. Each targets a group based on creating a good match to the unique performance, runtime, and price requirements of organizations and individuals. The five editions of SQL Server 2005 are as follows:

- Microsoft SQL Server 2005 Enterprise Edition
- Microsoft SQL Server 2005 Standard Edition

- Microsoft SQL Server 2005 Workgroup Edition
- Microsoft SQL Server 2005 Developer Edition
- Microsoft SQL Server 2005 Express Edition

The most common editions used are the Enterprise, Standard, and Workgroup editions as these work best in production server environments.

SQL Server 2005 Enterprise Edition (32-bit and 64-bit) This edition comes in 32- and 64-bit varieties. It's the ideal edition if you need to have a SQL Server 2005 that can scale to limitless size while supporting enterprise-sized online transaction processing (OLTP), highly complex data analysis, data warehousing systems, and websites.

In simplest terms, Enterprise Edition has all the bells and whistles and is beautifully suited to provide comprehensive Business Intelligence and analytics capabilities. It includes high-availability features such as failover clustering and database mirroring. It's ideal for large organizations or situations with the need for a SQL Server 2005 that can handle complex situations.

SQL Server 2005 Standard Edition (32-bit and 64-bit) Standard Edition includes the essential functionality needed for e-commerce, data warehousing, and line-of-business solutions without some advanced features such as Advanced Data Transforms, Data-Driven Subscriptions, and DataFlow Integration using Integration Services. Standard Edition is best suited for the small- to medium-sized organization that needs a complete data-management and analysis platform without many of the advanced features found in Enterprise Edition.

SQL Server 2005 Workgroup Edition (32-bit only) Workgroup Edition is the data-management solution for small organizations that need a database with no limits on size or number of users. It includes only the core database features of the product line (it doesn't include Analysis Services or Integration Services, for example). It's intended as an entry-level database that's easy to manage.

SQL Server 2005 Developer Edition (32-bit and 64-bit) Developer Edition has all the features of Enterprise Edition. However, it's licensed for use only as a development and test system, not as a production server. This edition is good choice for people or organizations who build and test applications but don't want to pay for Enterprise Edition.

SQL Server 2005 Express Edition (32-bit only) SQL Server Express is a free database that's easy to use and simple to manage. It comes without many of the features of other editions, including Management Studio, Notification Services, Analysis Service, Integration Services, and Report Builder, to name only a few. SQL Server Express can function as the client database or as a basic server database. It's a good option when all you need is a stripped-down version of SQL Server 2005, typically among low-end server users (such as small businesses), nonprofessional developers building web applications, and hobbyists building client applications. For more information about Express Edition, see *Mastering SQL Server 2005 Express Edition* by Mike Gunderloy and Susan Harkins (Sybex, 2006).

For more information on the differences in the various editions of SQL Server 2005, visit the SQL Server section of the Microsoft website at `http://www.microsoft.com/sql/2005/productinfo/sql2005features.mspx`.

Administering SQL Server

One of the key elements built into SQL Server 2005 is a high degree of integration between the development side and the management side. Microsoft designers worked to break down silos between the two groups and to make it possible to work together.

If you have experience with SQL Server 2000, the first thing you'll notice is that the Enterprise Manager has been completely revamped into SQL Server Management Studio, which performs most of the functions of Enterprise Manager along with many new ones.

We'll take a few moments to look at how you use this interface to administer data and services as well as to keep track of what's happening on your server. This will be a short visit, though. Management Studio is covered again briefly in Chapter 3. In addition, all of Chapter 9 is devoted to this important tool.

Opening SQL Server Management Studio

To launch Management Studio, choose Programs ➢ Microsoft SQL Server ➢ SQL Server Management Studio from the Windows Start menu. Management Studio is installed when you install SQL Server 2005. When it opens, you're asked to connect to a SQL Server instance. As soon as you connect, the SQL Server instance will appear in Object Explorer.

From here, you can expand a treeview to drill down from servers to databases to objects and inspect individual objects in a list view. Figure 1.1 shows how Management Studio might look after you drill down a few levels. In this case, we're examining the tables in the AdventureWorks database on a server named GARAK in the default server group.

NOTE The AdventureWorks database comes with SQL Server 2005. In many cases throughout the book, we use AdventureWorks as a good generic example of a database. We also use the AdventureWorks sample database or create examples that you can emulate for your own database needs.

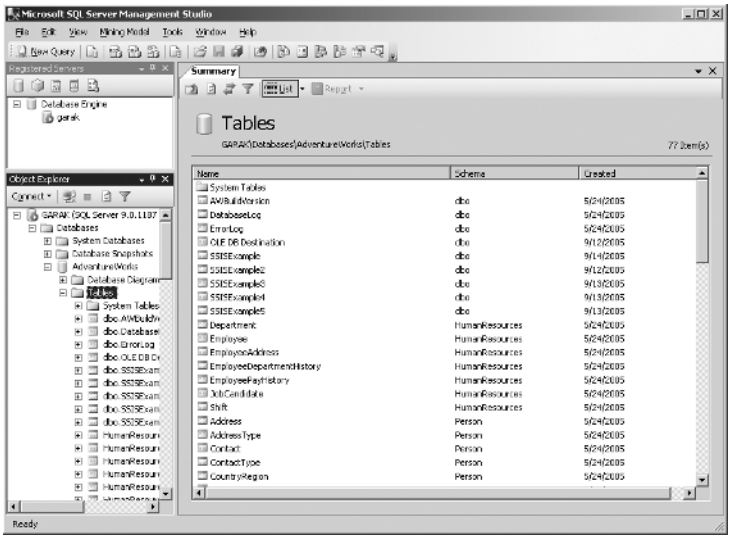


FIGURE 1.1
SQL Server Management Studio

Even if you don't know anything about Management Studio, you'll appreciate the wide list of objects that can be manipulated using this interface:

Databases	Alerts
Database diagrams	Operators
Tables	Jobs
Views	Backups
Stored procedures	Process information
Users	Database maintenance plans
Roles	SQL Server logs
Rules	Replication
Defaults	Logins
User-defined datatypes	Server roles
User-defined functions	Performance Analyzer
Full-text catalogs	Analysis services cubes
Integration Services packages	Linked servers
Metadata Services packages	Remote servers
Data Transformation Services metadata	Notification services subscriptions
Credentials	Shared schedules

And that's just a sample! You'll learn about most of these objects in coming chapters.

Creating a Login

If you're a DBA, one of your main tasks is managing security on your SQL Server. We'll discuss security in much more detail in Chapter 18, but for now, let's look at one part of the picture: creating a login. A SQL Server login is a necessary part of making your SQL Server data available to a Windows user on your network.

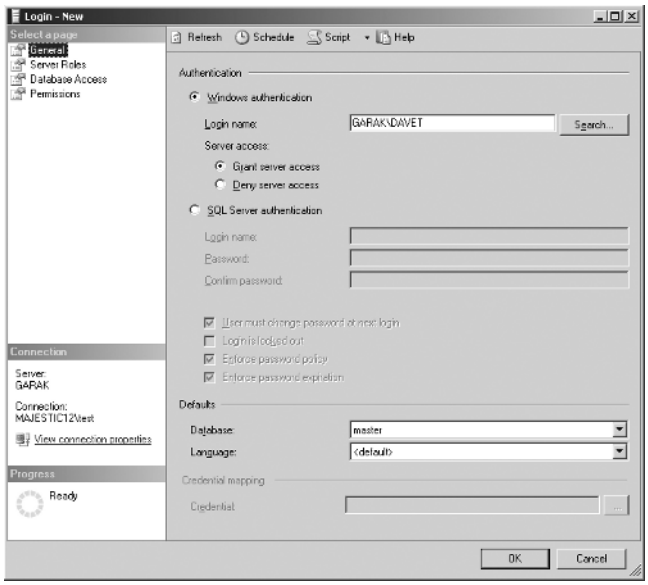
There are several ways to create a new login. The easiest technique is to use the Logins folder under the Security folder in Management Studio. Open Object Explorer, expand the server instance, and then expand the Security Folder. Right-click the Login folder, and select New Login. The Login - New property window opens (see Figure 1.2).

In the top part of the pane, you select the authentication mode. SQL Server can use two different methods to verify that a user is who they claim to be:

- Windows Authentication compares the user with their credentials in the Windows 2000/2003 user database.
- SQL Server Authentication prompts the user for a password that's evaluated by SQL Server itself.

In most cases, you should choose Windows Authentication—your users won't have to supply a separate password for SQL Server, and you won't have two sets of passwords to audit and coordinate. You might want SQL Server accounts, though, for operations such as accessing a database over the Internet. Also, you should be aware that Windows Authentication is available only if this copy of SQL Server is running on Windows 2000 or Windows 2003.

FIGURE 1.2
Login - New window, General page

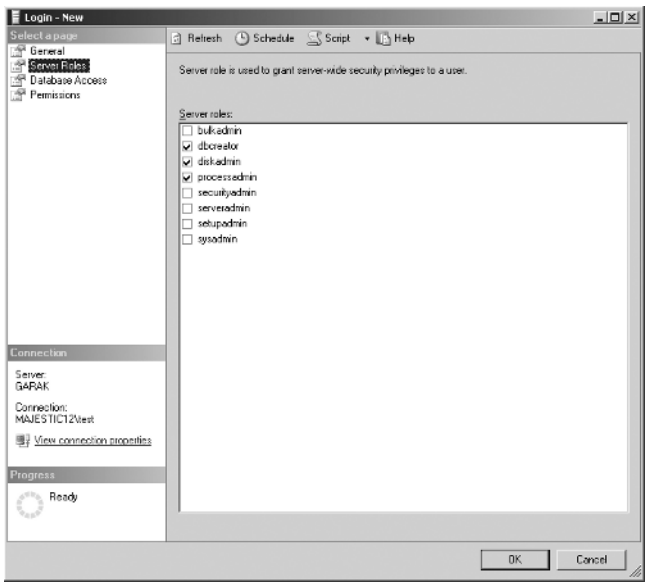


In the Login Name text box, you specify the Windows user for whom you want to create a login (assuming that you chose Windows Authentication mode). You can type in the domain and username manually or search for the user by clicking the Search button.

In the Server Access section, you can either grant a user access to your server or deny a user all access to your server. As a general rule, you should deny access to everyone who doesn't explicitly need to get to the data on your server. There's no point in having idle hands rifling through your database.

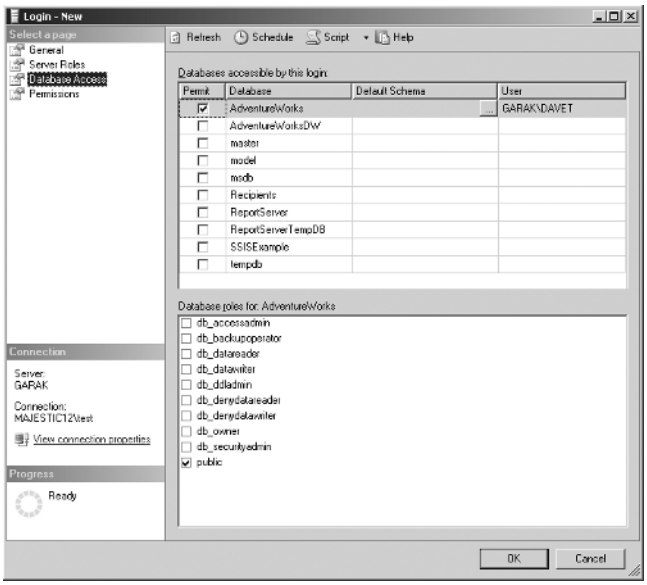
Now, click Server Roles in the left-hand pane to open the Server Roles page, as shown in Figure 1.3. Here you can select which server-wide security privileges this user should have.

FIGURE 1.3
Login - New window, Server Roles page



Select Database Access in the left-hand pane to open the Database Access page shown in Figure 1.4, and select what databases will be accessible to the specified login. If you don't choose any databases, the user can log in but can't do anything. If you want to set specific permissions, select Permissions in the left-hand pane, and select and set specific permissions.

FIGURE 1.4
Login - New window, Database Access page



If all is well, click OK to create the login. That's all there is to it!

Using the Configuration Manager

Another task you may be called on to perform is to change the way a SQL Server instance starts or make an adjustment to the SQL Server configuration. To do these sorts of tasks, you use a different server tool: the SQL Server Configuration Manager. To launch the SQL Server Configuration Manager, click Windows Start ➢ Programs ➢ Microsoft SQL Server ➢ Configuration Tools ➢ SQL Server Configuration Manager.

The SQL Server Configuration Manager opens as a Microsoft Management Console (MMC) snap-in. As you can see in Figure 1.5, it includes a number of trees.

Let's use SQL Server Configuration Manager to set SQL Server to start automatically. Select SQL Server 2005 Services. Next, in the Details pane, right-click the name of the SQL Server instance you want to start automatically, and then click Properties.

In the SQL Server Properties dialog box, click the Service tab, and set Start Mode to Automatic as shown in Figure 1.6. (Because this is the default value, it should already be set to Automatic.) Click OK, and then close SQL Server Configuration Manager.

Viewing Current Activity

At times, you may want to know what's going on in your database. You can get a quick overview through Management Studio by selecting the Activity Monitor node in the Management

section of Object Explorer. Right-click, and select View Processes from the pop-up menu to open the Activity Monitor Process Info page. Figure 1.7 shows typical activity on a lightly loaded server.

You may find a process running here that you don't recognize. If so, double-clicking the process lets you see the last set of T-SQL commands that were submitted by that particular process. If you're still in the dark, you can send a message from Enterprise Manager directly to the user or computer from which the process originated.

Other nodes within Management Studio allow you to easily view current locks and detect deadlock situations that may be harming performance.

FIGURE 1.5
SQL Server Configuration Manager

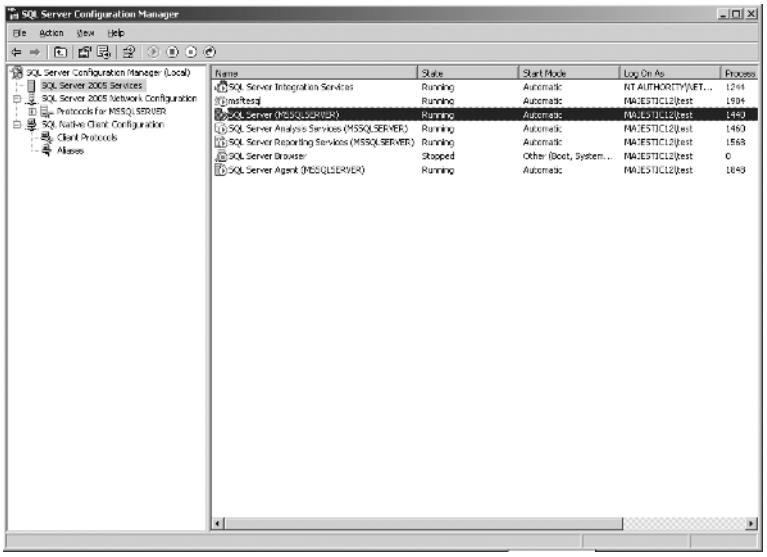


FIGURE 1.6
Setting Start Mode to Automatic

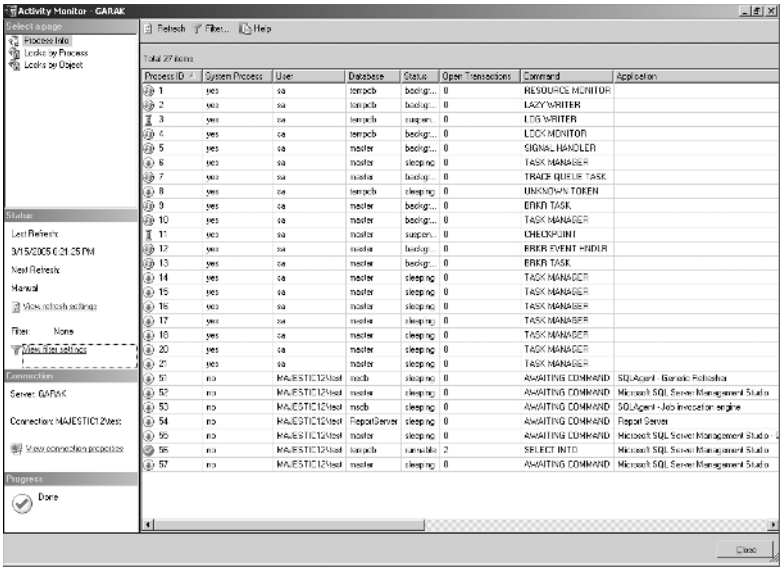


FIGURE 1.7
Viewing current activity

Development Tools

If you're a developer, you'll be less concerned with the design and maintenance of your database than with what you can do with it. SQL Server 2005 ships with a variety of tools for developers, including ActiveX Data Objects (ADO), SQL-DMO, SQL-NS, Integration Services, Analysis Services, and Bulk Copy Program (BCP). You'll learn about many of these in Parts 5 and 6 of this book, so we won't cover them in detail here. For now, we'll concentrate on one tool, Business Intelligence Development Studio (BIDS) and give you an overview of how much simpler design and development have become.

Business Intelligence Development Studio

Business Intelligence Development Studio (BIDS) is the SQL Server 2005 studio environment for developing Business Intelligence solutions including cubes, data sources, data source views, reports, and Integration Services packages.

To open BIDS, click Windows Start ➢ Programs ➢ Microsoft SQL Server ➢ SQL Server Business Intelligence Development Studio. You'll be using BIDS to work on projects, so click File ➢ New ➢ Project to open the New Project window, shown in Figure 1.8.

Note that there are six projects or wizards covering three different technologies to choose from: Analysis Services Project, Import Analysis Services 9.0 Database, Integration Services Project, Report Server Project Wizard, Report Model Project, and Report Server Project.

For now, let's look at a the AdventureWorks Sample Reports in BIDS, as shown in Figure 1.9. (If you want, you can open the project yourself— assuming you've installed the sample— by clicking Open ➢ Project Solution then navigating to `C:\Program Files\Microsoft SQL Server\90\SamplesReporting Services\Report Samples\AdventureWorks Sample Reports\ AdventureWorks Sample Reports.sln` and then selecting the Sales Order Detail.rdl item in the Reports folder.)

FIGURE 1.8
BIDS New Project window

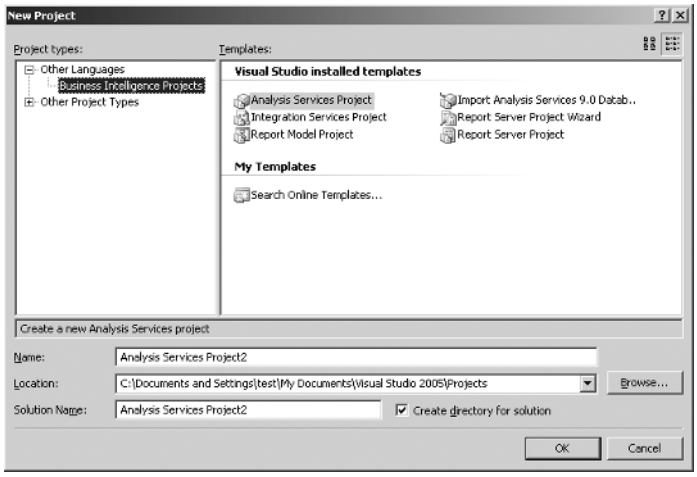
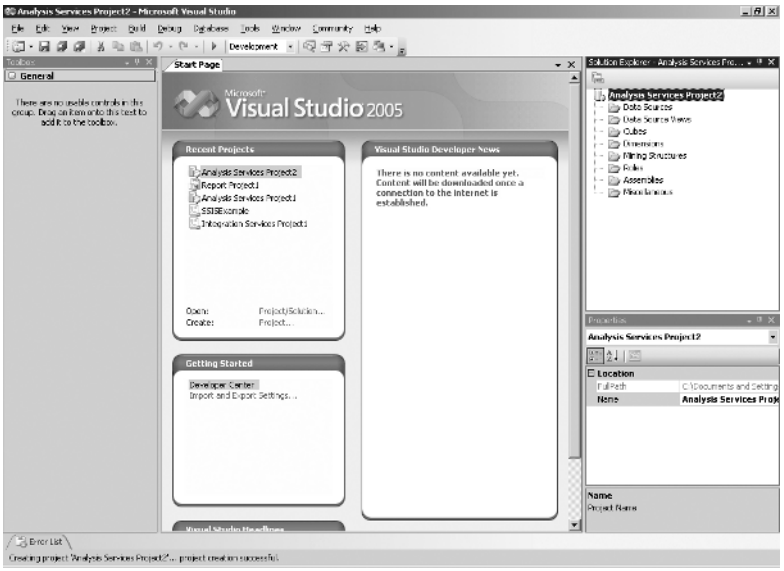


FIGURE 1.9
Sample Project in Business Intelligence Development Studio



As you can see, the BIDS project consists of four main windows:

Designer window The Designer window, which appears as the central pane with Toolbox to the left and the Solution Explorer and Properties panes to the right, provides a graphical view of an object and is used to create and modify Business Intelligence objects. Each of the three SQL Server 2005 Business Intelligence components has a designer tailored to it. For example, the Integration Services Package Designer provides the design surfaces to create Integration Services packages, and the Report Designer does the same when you create and preview reports. Some object types, such as data source views, are available to all Business Intelligence projects, and Data Source View Designer is included in all project types. The designers provide a code view and a design view of an object.

Solution Explorer Solution Explorer provides you with an organized view of your project and associated files as well as easy access to the commands relevant to them. A toolbar in this window offers commonly used commands for the item you highlight in the list.

Properties window The Properties window is used to view and change the design-time properties and events of selected objects in editors and designers. This window displays different types of editing fields including edit boxes, drop-down lists, and links to custom editor dialog boxes. Properties shown in gray are read-only.

Toolbox window This window shows a variety of items for use in Business Intelligence projects. The tabs and items available from the Toolbox change, depending on the designer or editor currently in use. The Toolbox always displays the General tab. Additional tabs may display depending on the project type.

BIDS also includes other windows for viewing search results and error and output information. Windows and their contents change depending on the type of project you're working on.

All BIDS projects are developed within a *solution*. A solution is a server-independent container that can include multiple Integration Services projects as well as Analysis Services and report projects.

You'll learn more about BIDS and its use as a powerful development tool in Chapters 22, 26, and 28 when we cover Integration Services, Analysis Services, and Reporting Services.

New and Improved

A great deal has changed in SQL Server 2005, and although you may be familiar with earlier versions, what you've seen so far should be enough to convince you that things in SQL Server are very different. The depth and flexibility of the product as reflected in the new Management Studio and BIDS should be enough to convince you that knowledge of SQL Server 2000 isn't enough to truly master SQL Server 2005.

To attempt to list and explain every change, new feature, and enhancement in SQL Server 2005 would likely take another book. In the next few pages, we'll introduce those that you're likely to see in the components you'll probably make the most use of.

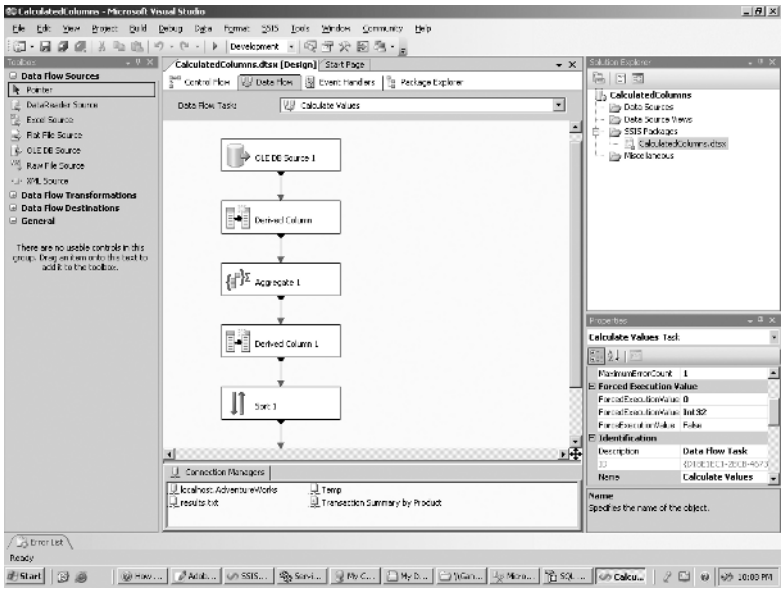
Integration Services

SQL Server Integration Services (SSIS) is virtually a complete redo of the old Data Transformations Services, as you'll see in Chapter 22. With such wholesale changes, we can touch on only a few of them here. Key differences include the introduction of graphical tools, such as the SSIS Designer through BIDS (shown in Figure 1.10) and the SQL Server Import and Export Wizard; increased extensibility by using custom tasks, sources, destinations, and transformations; and changes to the architecture.

Data flow and control flow have been separated into two distinct engines—the Integration Services runtime engine and the Integration Services data flow engine. This separation provides better control of package execution, increases the visibility of data transformations, and enhances the extensibility of Integration Services by simplifying the creation and implementation of custom tasks and transformations.

Support for the Microsoft .NET Framework makes it easier to create custom Integration Services tasks, transformations, and data adapters.

FIGURE 1.10
Integration Services Package Designer



NEW TASKS

The following new tasks have been added:

- The WMI Data Reader task for querying Windows Management Instrumentation (WMI) data
- The WMI Event Watcher task for listening to WMI events
- The File System task for performing operations on files and folders in the file system
- The Web Service task for accessing web services
- The XML task for working with XML documents
- The Analysis Services Execute DDL task for running DDL scripts
- The Data Mining Query task for querying data mining models

NEW DATA SOURCES AND DESTINATIONS

SSIS now uses the following new sources and destinations in addition to the SQL Server, OLE DB, and flat file sources and destinations:

- Data Mining Query destination
- DataReader source and destination
- Dimension Processing destination
- Partition Processing destination
- Raw File source and destination

- Recordset destination
- SQL Server Mobile destination
- The Data Mining Model Training destination
- XML source

Integration Services also includes the Script Component for simplified development of custom sources and destinations.

New Data Transformations

There are 20 new transformations, described in detail in Chapter 22, making it easier for developers to build packages with complex data flow without writing any code.

Managing and monitoring packages is now easier with the following new tools:

Integration Services service The Integration Services service is a new Microsoft Windows service that manages package storage and displays a hierarchical view of saved packages in Management Studio. The service supports packages stored in the msdb database in an instance of SQL Server or in the file system.

DTUTIL The `dtutil` command-line utility lets you copy, delete, move, and sign packages stored in the msdb database, in an instance of SQL Server, or in the file system.

Running packages list This tool displays a list of running packages in Management Studio.

Package logging options SSIS includes multiple logging providers, a logging schema from which you can choose the type of information to log, and a flexible logging model that supports logging configuration at the package level and the task level.

Package restart capability Checkpoints can be set to let you restart a package from the failed task instead of having to rerun the whole package.

Security features New security features include:

- Roles can be used for packages in the msdb database in an instance of SQL Server.
- Packages can be encrypted with various levels of encryption to protect sensitive data.
- Packages can now be digitally signed.

New and updated Integration Services wizards Integration Services includes a set of new and updated wizards that help you accomplish complex tasks such as deploying, importing, and exporting packages or migrating SQL Server 2000 Data Transformation Services (DTS) packages from the SQL Server 2000 format to SQL Server 2005, as well as other tasks as listed in Table 1.1.

Table 1.1: Integration Services Wizards

Wizard	Description
SQL Server Import and Export Wizard	Creates packages that copy data between a source and a destination
Package Configuration Wizard	Creates configurations that can be deployed with packages

TABLE 1.1: Integration Services Wizards *(CONTINUED)*

WIZARD	DESCRIPTION
Package Installer Wizard	Deploys packages and updated package configurations
Package Migration Wizard	Migrates SQL Server 2000 DTS packages to SQL Server 2005 Integration Services packages

Replication

Replication is the process of copying and distributing data and database objects from one database to another and then synchronizing between databases to maintain consistency. Without a good replication modality in place, you'll get in trouble sooner or later if more than one person is accessing the database, or if more than one copy of it exists.

Take this simple example: Imagine you have a database of all your DVDs on your home network. You copy the database onto your laptop and go on vacation. Three days into the vacation, you buy more DVDs and enter them into the database on your laptop. However, what you don't know is that the day after you left, your son purchased a new DVD and added it to the database at home. You return home and copy the laptop database over to your home database. It's now "updated" with the most recent files. Unfortunately, it's wrong, because you've erased all entries except those made by you. DVDs are one thing—imagine what it would be like if those were your financial records, or a company's inventory database. If you didn't have replication, then eventually drift would occur; instead of one database, you'd have many, each containing slightly different data.

Using replication, you can distribute data to different locations and to remote or mobile users over local and wide area networks, dial-up connections, wireless connections, and the Internet.

Recognizing the important role replication plays in good database health, several new features and enhancements have been added in SQL Server 2005. As you'll see in Chapter 25, a number of enhancements make keeping databases properly synchronized and replicated much easier, even with non-SQL Server 2005 databases:

Simplified user interface The New Publication Wizard has 40 percent fewer pages than its SQL Server 2000 counterpart.

The Push Subscription Wizard and Pull Subscription Wizard are combined into the New Subscription Wizard, which provides a convenient way to create multiple subscriptions with different properties.

Replication Monitor has been completely redesigned (see Figure 1.11).

Improved replication between different types of databases You can now publish from Oracle databases to SQL Server 2005. In addition, you can now publish data to Oracle and IBM DB2, using snapshot and transactional replication.

Replication Management Objects (RMO) RMO is a Microsoft .NET library that provides a set of common language runtime classes for configuring, managing, and scripting replication, and for synchronizing Subscribers. This means programs can work on individual classes, such as the publication or subscription class, without traversing from top-level classes.

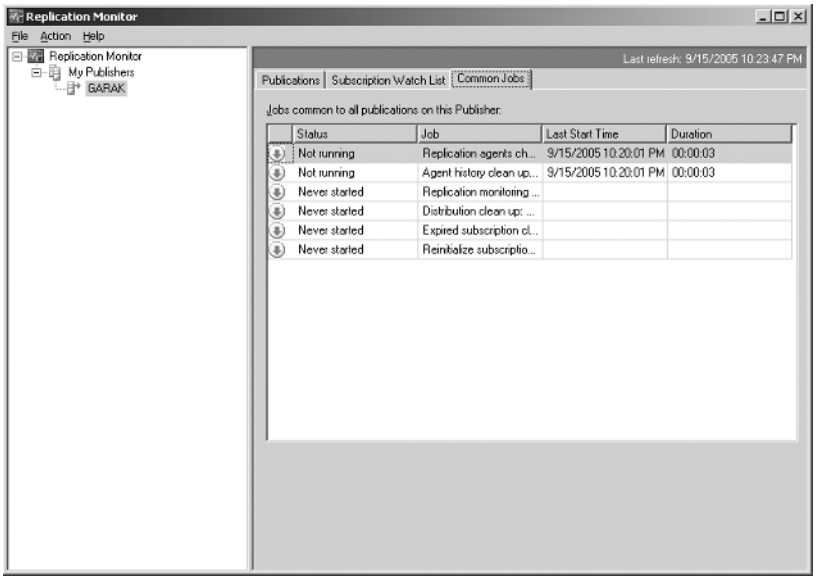


FIGURE 1.11
Replication Monitor has been completely redesigned for SQL Server 2005

Business logic handlers for merge replication The business logic handler framework allows you to write a managed code assembly that is called during the merge synchronization. This means that if a salesperson enters an order from a handheld device, the inventory can be checked during synchronization, and the salesperson can be notified if the product is sold out.

Changes to transactional replication

- Peer-to-peer transactional replication
- Ability to initialize a transactional subscription from a backup
- Ability to modify call formats for transactional articles without reinitializing
- Increased number of columns allowed in transactional publications
- Tracer tokens for transactional publications
- Default use of concurrent snapshots by transactional publications

Other enhancements In addition to those already listed, SQL Server 2005 adds the following features to its replication capabilities:

- Improved identity range management
- Parallel snapshot preparation
- Resumable snapshot delivery
- Improved monitoring statistics for merge subscriptions
- Improvements to snapshots for merge publications with parameterized filters
- Declarative ordering for articles in merge publications
- Conditional delete processing for articles in merge publications
- Replication of schema changes

- Replication of logical records
- Improved error messages

Analysis Services

Many changes have been made to Analysis Services, sometimes still referred to as *business analytics*. As you'll see in Chapter 26, the topic of Analysis Services is vast.

Building on its beginnings in SQL Server 2000, Microsoft SQL Server 2005 Analysis Services (SSAS) provides additional support for Business Intelligence, delivering increased scalability, availability, and security to Business Intelligence solutions while making them easier to create, deploy, and manage. Many new features, and improvements to existing features, have been added to Analysis Services as shown in the following tables.

SSAS comes with a number of new and advanced designers, as described in Table 1.2

TABLE 1.2: New and Improved SSAS Designers

DESIGNER	DESCRIPTION
Cube Designer	Now provides support for dimension usage, translation, MDX scripting, and Key Performance Indicator (KPI) functions
Data Mining Model Designer	Used for defining, viewing, and testing mining structures and mining models in BIDS
Data Source View Designer	Provides a simple, diagram-based environment for defining the tables and relationships in a Data Source view on which to base Analysis Services objects
Dimension Designer	Enhanced to provide support for attribute-based dimension definitions, user-defined and attribute hierarchies, translations, and dimension writeback

New and improved wizards make many SSAS tasks much simpler to perform for users, developers, and DBAs. Table 1.3 summarizes these wizards.

TABLE 1.3: New and Improved SSAS Wizards

WIZARD	DESCRIPTION
Business Intelligence Wizard	Provides advanced Business Intelligence features, such as currency conversion, account intelligence, time intelligence, and dimension writeback support.
Cube Wizard	Walks you through the steps of designing and prototyping a cube. Provides several enhancements, including autobuild technology, to analyze and determine dimensions and hierarchies intelligently and to measure groups from the tables and relationships of the underlying data source (see Figure 1.12).
Data Mining Model Wizard	Creates new mining structures based on either relational or multidimensional data that can be modified later by using the Data Mining Designer.

TABLE 1.3: New and Improved SSAS Wizards *(CONTINUED)*

WIZARD	DESCRIPTION
Data Source View Wizard	Quickly and automatically retrieves the relational schema of a data source and constructs tables and relationships on which Analysis Services objects, such as dimensions and cubes, can be based.
Dimension Wizard	Adds slowly changing dimension support, dimension writeback, account intelligence, and time intelligence to the design of database dimensions in Analysis Services.
Migration Wizard	Migrates databases from previous versions of Analysis Services to an instance of SSAS.
Schema Generation Wizard	Allows you to create relational schemas based on existing Analysis Services objects. Can be used to define your dimensions and cubes first and then design a Data Source view based on your dimensions and cubes. The Data Source view can be used to create and populate a relational database specifically to support your Business Intelligence solution.

FIGURE 1.12
Cube Wizard

SSAS' Analysis Services service has been beefed up and expanded to include the changes and enhancements described in Table 1.4.

TABLE 1.4: SSAS Changes and Improvements

FEATURE	DESCRIPTION
Failover clustering	Support for 8-node failover clusters on 32-bit systems and 4-node clusters on 64-bit systems.
Language and collation support	Support for language and collation settings at both the instance level and the database level.
Multi-instance support	Ability to install up to 50 instances of the Analysis Services service from Microsoft SQL Server 2005 Enterprise Edition on one computer. Up to 16 instances of the Analysis Services service can be installed from other editions of SQL Server 2005.
Orphan fact table rows	Ability to use settings for each hierarchy in a dimension to determine how to handle orphan fact table rows.
Proactive caching	Used to increase the performance of dimensions, partitions, and aggregations.
Processing support	Additional flexibility, including direct support of parallel processing.
Scripting support	Ability to script databases and subordinate objects by using the Analysis Services Scripting Language (ASSL).
XML support	Full implementation of the XML for Analysis (XMLA) 1.1 specification.

Table 1.5 shows the SSAS management enhancements and new features available in SQL Server 2005.

TABLE 1.5: SSAS Management Enhancements

FEATURE	DESCRIPTION
Deployment engine	Analysis Services now includes its own engine for deploying Analysis Services projects and solutions.
Security	Analysis Services provides increased security features, including better access control, encryption, and monitoring tools.
SQL Server Profiler integration	Analysis Services now supports SQL Server Profiler for monitoring and capturing any events that are generated by an instance of Analysis Services for future analysis or playback.

SSAS includes the cube, dimension, and data-mining enhancements and new features shown in Table 1.6.

TABLE 1.6: SSAS Cube, Dimension, and Data-Mining Enhancements

TOPIC	ITEM	DESCRIPTION
Cube	Key Performance Indicators	KPIs are customizable business metrics used by companies to track performance and improve performance.
Cube	Multiple fact tables	Multiple fact tables within a single cube are supported through the use of measure groups.
Cube	Perspectives	New perspectives let you define a viewable subset of the cube and can provide a focused, business-specific or application-specific viewpoint on a cube.
Cube	Semi-additive measures	Semi-additive measures enable aggregation for an account dimension to be set by account. Business users can then set up cubes that reflect a company's account structure without writing custom rollup formulas.
Dimensions	Attributes	Dimensions are now based on attributes, which correspond to the columns in the tables of a dimension. Each attribute contains the members of a dimension table column.
Dimensions	Linked measure groups and dimensions	Data from different data sources can be used by linking a cube to a measure group in another cube that is stored either in the same database or in a different database on an instance of SSAS. You can also link a cube to a dimension in another database.
Dimensions	Multiple hierarchies	Multiple hierarchies are supported in a single dimension.
Dimensions	Simplified dimension types	Two dimension types, standard and linked, replace the four dimension varieties in SQL Server 2000 Analysis Services.
Data mining	Microsoft Association Algorithm	This algorithm builds rules that describe which items are most likely to appear together in a transaction.
Data mining	Microsoft Linear Regression Algorithm	This algorithm provides linear regression support.
Data mining	Microsoft Logistic Regression Algorithm	This algorithm provides logistic regression support.
Data mining	Microsoft Naive Bayes Algorithm	This algorithm is used to explore data between input columns and predictable columns and discover the relationships between them.

TABLE 1.6: SSAS Cube, Dimension, and Data-Mining Enhancements *(CONTINUED)*

TOPIC	ITEM	DESCRIPTION
Data mining	Microsoft Neural Network Algorithm	This algorithm creates classification and regression mining models by constructing a multilayer perceptron network of neurons. Ideal for nonlinear models.
Data mining	Microsoft Sequence Clustering Algorithm	This algorithm identifies clusters of similarly ordered events in a sequence that can be used to predict the likely ordering of events in a sequence based on known characteristics.
Data mining	Microsoft Time Series Algorithm	This algorithm analyzes time-related data, such as monthly sales data or yearly profits, for patterns to use to predict values for future time steps.

Other changes to cubes, dimensions, and data mining include the following:

- Data and metadata are now loaded into memory only when needed, allowing dimensions of virtually unlimited size.
- Several tasks have been added to SSIS that can be used to create a complete data-mining solution.
- Member group requirements for dimensions have been eliminated.
- Support is now provided for
  - Fact dimensions through Fact Dimensions Relationships
  - Many-to-many relationships between fact tables and dimension tables by using association tables
  - Reference dimensions through the use of Reference Dimension Relationships, in which a reference dimension is indirectly coupled to a measure group by another dimension
  - Role Playing Dimension Relationships, which express multiple relationships between a dimension table and a fact table as a single dimension.

Developers and programmers will be pleased to see that they have not been forgotten. SSAS introduces the development enhancements and new features summarized in Table 1.8.

TABLE 1.7: SSAS Development Changes and improvements

FEATURE	DESCRIPTION
ADOMD.NET	Formerly part of the SQL Server 2000 ADOMD.NET SDK, ADOMD.NET is now fully integrated into SSAS.
Analysis Management Objects (AMO)	AMO replaces the Decision Support Objects (DSO) object model.

TABLE 1.7: SSAS Development Changes and improvements *(CONTINUED)*

FEATURE	DESCRIPTION
Microsoft .NET Framework support	SSAS is now fully integrated with the Microsoft .NET Framework.
Multidimensional Expressions enhancements	Multidimensional Expressions (MDX) language has added support for scripting, scope and context control, and enhanced subcube manipulation.
Persisted calculations	The results of calculated members or calculated cells of cubes can now be persisted and managed in a separate cache for each cube.
Stored procedures	SSAS provides more extensibility and programmability in stored procedures, external routines in programming languages such as C#, C++, or Visual Basic that you can use to extend SSAS functionality.

Notification Services

Notification Services is a new platform for developing and deploying applications that generate and send notifications. Notification Services can send timely, personalized messages to thousands or millions of subscribers using a wide variety of devices. For example, you can use Notification Services to send a text message to a cell phone about a stock price when a certain price is reached.

Notification Services 2.0 was a downloadable component of SQL Server 2000 and was released in 2002. In SQL Server 2005, Notification Services is integrated into SQL Server.

For DBAs, the main advantage is that Notification Services is now fully integrated into Management Studio. Using Object Explorer, you can perform most of the tasks you previously needed to do at the command prompt using the `NSCONTROL` utility. You can start and stop instances of Notification Services.

If you're a developer, you can use Management Studio as your XML and T-SQL editor for a Notification Services instance. You can easily edit your instance configuration file (ICF), application definition files (ADFs), and T-SQL scripts for managing security or administering the instance, and you can then deploy the instance using Object Explorer.

In Notification Services 2.0, an application developer defined the complete T-SQL action for generating notifications, and subscribers could only provide parameters for the action. Now Notification Services has a new type of action, the *condition action*. Subscribers can now fully define their own subscriptions over the data set.

SQL Server Notification Services has a new management API, `Microsoft.SqlServer.Management.Nmo`. You can use this API to develop and manage Notification Services instances and applications.

Notification Services has added a new standard event provider to gather event data from SSAS databases using MDX queries.

The following views have been added or modified to simplify application development and troubleshooting:

- *<EventClassName>*—One of these is created for each event class defined in an application. When you write event-driven (not scheduled) notification generation queries, you typically select events from this view. Now you also can insert event data into this view.
- *<NotificationClassName>*—One of these is created for each notification class. You can use this view to review notifications generated by your application.

Notification Services now provides three views for viewing and managing subscriber and subscription data:

- NSSubscriberView lists all of the subscribers for an instance of Notification Services. You can use this view to manage subscriber data.
- NSSubscriberDeviceView lists all of the subscriber devices for an instance of Notification Services. You can use this view to manage subscriber device data.
- NS*SubscriptionClassName*View lists all the subscriptions for a subscription class. You can use this view to manage basic event-driven subscriptions but not scheduled or condition-based subscriptions.

You'll learn a lot more about Notification Services in Chapter 27.

Reporting Services

If there is one technology that will be heavily used by everyone who accesses a SQL Server 2005 instance, then the strongest candidate is Reporting Services.

Having well programmed, beautifully managed, exquisitely replicated, exceptionally developed, and brilliantly analyzed data means absolutely nothing if you can't get it out of the database in a form that anyone can use. For most users, reporting is still the heart of database management.

SQL Server 2000 didn't originally ship with a Reporting Services component. However, in 2002, under growing pressure from the SQL Server community, Microsoft released Reporting Services as a free add-on that was downloaded by hundreds of thousands of people.

Although there have been many improvements to Reporting Services, without a doubt those with the most far-reaching impact are the new Report Builder, the new Model Builder, and very enhanced Report Designer, which is fully integrated with BIDS:

Report Builder Report Builder may prove to be the hottest addition in SQL Server 2005. Designed to be used by end users without extensive technical knowledge, it's used from a web-based interface to generate ad hoc reports. Report Builder can be accessed through a URL or from Report Manager.

Model Builder A new type of project, Report Model, has been added. Report Models are used by Report Builder to generate ad hoc reports. You create a model using Model Designer in BIDS. Model Designer has several wizards to help you specify data sources and data views and generate models.

Report Designer The new Report Designer runs in BIDS and contains a number of changes and improvements from its predecessor:

- Expression Editor now includes functions for authors as well as Intellisense features.
- You can now specify data sources dynamically, permitting you to switch data sources at runtime based on conditions you specify in the expression.
- A new Analysis Services Query Designer helps you create MDX queries.
- A new data-processing extension allows you to build reports from data generated by an SSIS package.

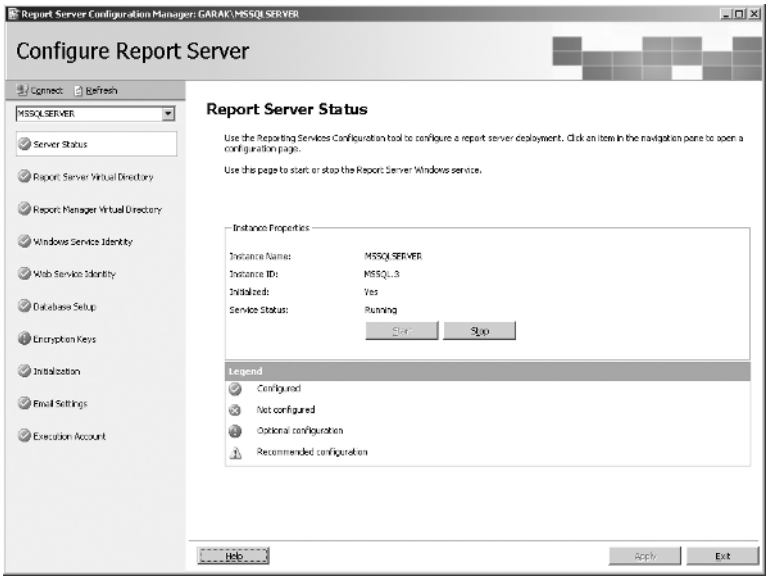
Report functionality SQL Server 2005 Reporting Services includes several improvements in report functionality of particular benefit to users:

- Interactive sorting in reports
- Ability to print multipage reports
- Ability to use multivalued parameters

Reporting Services Configuration Tool This new tool runs from the Start menu on the computer that hosts the report server. It can be used to configure a report server to create and use a report server database on a remote SQL Server instance, among other things, as you can see in Figure 1.13. You can also use this tool to specify accounts for the Microsoft Windows and web services, virtual directories, and e-mail delivery. Deploying multiple report servers on cluster nodes (previously known as a *report server web farm*) is now handled exclusively through the Configuration Tool or through configuration scripts.

This is of course only the tip of the iceberg, as you'll see when you learn more about Reporting Services in Chapter 28.

FIGURE 1.13
Reporting Services Configuration Tool



Service Broker

SQL Server 2005 introduces a completely new technology called Service Broker. The role of Service Broker is to aid in the building of database-intensive distributed applications that are secure, reliable, and scalable.

Part of the database engine, Service Broker provides facilities for storing message queues in SQL Server databases. In addition, Service Broker provides new T-SQL statements used by applications to send and receive messages. Each message is part of a *dialog*: a reliable, persistent communication channel between two participants.

Other things that Service Broker brings to SQL Server 2005 are as follows:

- An asynchronous programming model that allows database applications to perform tasks as resources become available
- Reliable messaging between SQL Server instances using TCP/IP
- A consistent programming model that can be used for messages whether they're within the same instance or between many instances, improving the ability to scale up or scale down by applications

Because Service Broker is elegantly designed to implement messaging within the SQL Server database engine, message queues are part of SQL Server databases and can take advantage of the performance capabilities of the database engine. In addition, Service Broker automatically handles issues such as message ordering and grouping, and a built-in locking capability allows only one reader at a time to read messages in a conversation group.

Finally, Service Broker stores message queues as part of the database. Hence, they are backed up and restored when the database is. Database security features can be used to secure applications. Similarly, messaging operations become an integral part of any transaction that includes database data, meaning there is no need to manage distributed transactions as you would have to do if the message queue were managed by a service separate from the database engine.

Service Broker is discussed in Chapter 29.

Summary

SQL Server isn't everything to everybody, but in the current release, it certainly has something for almost every computer user. The range of SQL Server goes from simple customer databases intended for a single user all the way to terabytes (a *terabyte* is one trillion characters) of data in cases such as Microsoft's TerraServer (`http://www.terraserver.microsoft.com`).

In the rest of this book, you'll learn about various aspects of SQL Server:

- Part 1 will teach you basic SQL Server and database concepts.
- Part 2 will teach you Transact-SQL.
- Part 3 examines the basic SQL Server objects and Management Studio.
- Part 4 covers administrative tasks.
- Part 5 reviews the developer tools that ship with SQL Server.
- Part 7 introduces some advanced topics and new technologies.

We hope that what you've read in this chapter has whetted your appetite for more. The next chapter will introduce you to some basic database concepts.