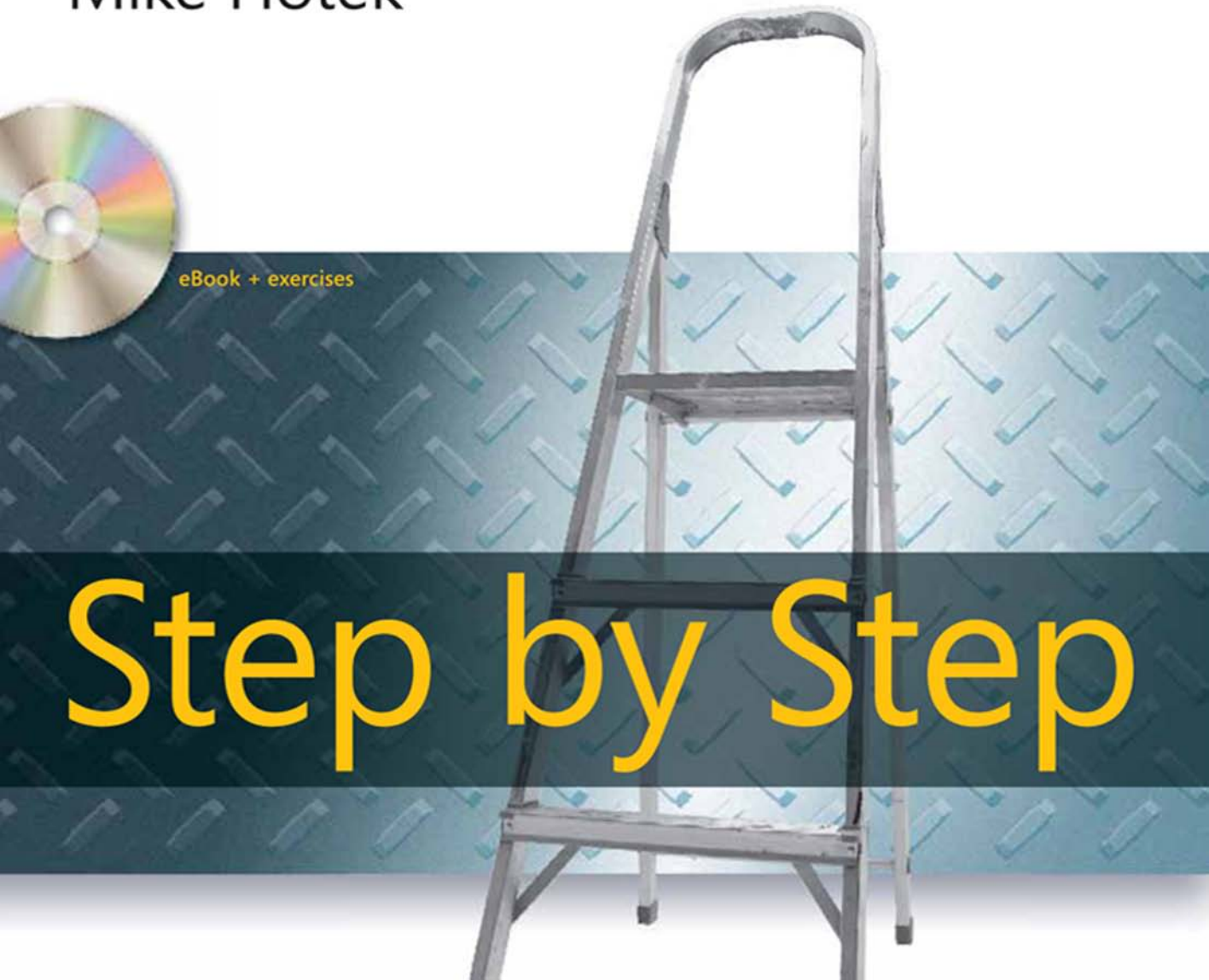
Microsoft

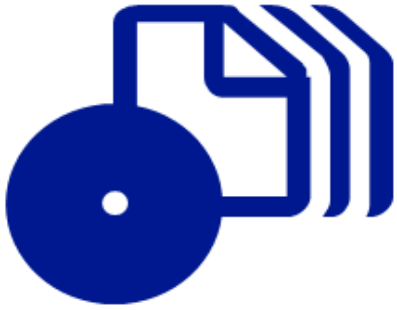
Microsoft® SQL Server® 2008

Mike Hotek



Step by Step

How to access your CD files



The print edition of this book includes a CD. To access the CD files, go to http://aka.ms/626041/files, and look for the Downloads tab.

Note: Use a desktop web browser, as files may not be accessible from all ereader devices.

Questions? Please contact: mspinput@microsoft.com

Microsoft Press

PUBLISHED BY
Microsoft Press
A Division of Microsoft Corporation
One Microsoft Way
Redmond, Washington 98052-6399

Library of Congress Control Number: 2008935428

Printed and bound in the United States of America.

ISBN: 978-0-7356-2604-1

5 6 7 8 9 10 11 12 13 QGT 7 6 5 4 3 2 1

Distributed in Canada by H.B. Fenn and Company Ltd.

A CIP catalogue record for this book is available from the British Library.

Microsoft Press books are available through booksellers and distributors worldwide. For further information about international editions, contact your local Microsoft Corporation office or contact Microsoft Press International directly at fax (425) 936-7329. Visit our Web site at www.microsoft.com/mspress. Send comments to mspinput@microsoft.com.

Acquisitions Editor: Ken Jones
Developmental Editor: Sally Stickney
Project Editor: Denise Bankaitis
Editorial Production: S4Carlisle Publishing Services
Technical Reviewer: Randall Galloway; Technical Review services provided by Content Master, a member of CM Group, Ltd.
Cover: Tom Draper Design

Body Part No. X15-12274

[2011-12-16]

To Genilyn,

You make it all worthwhile.

Contents at a Glance

Table of Contents

Part V Database Management

Acknowledgments

Thank you to all of my readers over the past decade or so; it's hard to believe that this will be the eighth book I've written and it would not be possible without you. I'd like to thank the talented and incredibly patient editorial team at Microsoft Press – Denise Bankaitis and Sally Stickney. I would especially like to thank Ken Jones, who has now gone through four books with me and has an exceptional talent for keeping things moving smoothly through all of the various trials that come up during the authoring and editing processes. My words and thoughts would probably be an unintelligible mess without the help of Randall and Christian, who not only smoothed out the rough edges, but made sense out of many 3 A.M. missives that likely had them questioning my sanity.

Introduction

Microsoft SQL Server has been Microsoft's flagship database for over 15 years. Before the next version of SQL Server launches, SQL Server 2008 will be celebrating SQL Server's 20th birthday. In that time, SQL Server has grown from handling small departmental tasks to serving up the largest databases on the planet. The release of SQL Server 2000 saw a dramatic evolution of Microsoft SQL Server. No longer a simple "database," Microsoft SQL Server is now a complete data architecture solution capable of handling the data storage and manipulation needs of any organization.

SQL Server 2005 extended the data platform with dramatic new capabilities in programming, .NET integration, high availability, management instrumentation, and business intelligence. So dramatic were the feature enhancements, that while retaining the same "look and feel" of Microsoft SQL Server, Microsoft essentially released and entirely new data platform.

SQL Server 2008 picks up the rapid innovation by enhancing hundreds of existing features while at the same time adding hundreds more. SQL Server 2008 also presents a highly scalable, highly adaptable data architecture platform against which you can build any conceivable application. Yet with these new and improved features come challenges for IT professionals. I know from nearly two decades of experience working with and teaching Microsoft SQL Server in organizations of all sizes spanning all industries that if users don't understand how to use the product effectively, they and their organization won't be able to get the full benefits of this powerful product. At the same, the role and skill set of a DBA is changing rapidly. While it might be acceptable for a consultant to focus on a very narrow area such as performance tuning, the storage engine, or writing reports, most companies are beginning to insist that their DBAs know how to architect, code, and manage solutions that utilize every feature that ships with SQL Server – from building a table through building a data mining model.

I wrote this book because I wanted to provide, the first comprehensive tour of the entire feature set available within Microsoft SQL Server, beginning with the relational databases that lie at the core of every organization, through the unique management capabilities, and finishing with a set of extraordinarily powerful analysis platforms which comprise the core of the Microsoft business intelligence platform. Armed with this information, you will be able to:

- Architect, secure, and manage relational databases
- Retrieve and manipulate data
- Expand your application's capabilities with programmable objects
- Secure and recover your business data
- Ensure that your database platform performs well and is resilient
- Transform your business data into actionable business intelligence

Who This Book Is For

The aim of this book is to teach you the fundamentals of the SQL Server 2008 data platform. SQL Server contains features that appeal to anyone involved with the storage or manipulation of data within an organization.

This book is intended for the IT professional who is either new to SQL Server or new to SQL Server 2008. Experienced SQL Server professionals will still find a significant amount of information that is applicable to their jobs.

No book can possibly include all of the intricacies of SQL Server 2008. Instead, the focus is on providing an overview of each feature in sufficient depth to allow you to build SQL Server applications. As you progress through this book, you will learn how to install each of the components, configure and manage instances, and build databases. You will walk through each of the client tools that ship with SQL Server, and I'll explain how each tool enables you to develop and manage your database environment. You will learn how to manipulate data, secure your databases, manage and protect your data, and distribute your data platform to make it more scalable, redundant, and fault tolerant. You will learn how to integrate your SQL Server data with a variety of sources, build reports to serve your lines of business, and finally feed all of this data into powerful analysis and data mining systems to deliver actionable information to your lines of business in near real-time.

How This Book Is Organized

This book is organized in seven parts along with three online articles that allow you to focus on specific portions within the SQL Server platform as well as specific job functions as follows:

- Part Onedescribes the core components available within SQL Server along with instructions on how to install each component. You will also get an overview of each tool that ships with SQL Server that we will be using throughout the book.
- Part Two shows you how to create and configure databases to provide the foundation for the table and index structures that form the backbone of every database application you will create.
- Part Three teaches you how to manipulate and retrieve data.
- Part Four introduces you to the fundamental programming structures, views, stored procedures, functions, and triggers available.
- Part Five explains how to secure, manage, backup, and recover databases.
- Part SixI will provide an introduction to the high availability technologies which ship with SQL Server such as clustering, database mirroring, and log shipping.
- Part Seven covers the three business intelligence technologies – Integration Services, Reporting Services, and Analysis Services.

- Bonus material is provided online in three articles, "Performance Analysis and Tuning", "Performance and Data Capture Tools", and "Performance Analysis Tools" on the Microsoft Press Online Windows Server and Client Web site at *www.microsoft.com/learning/books/online/serverclient*

Finding Your Best Starting Point in This Book

This book is designed to help you build skills in a number of essential areas. You can use this book if you are new to SQL Server or if you are switching from another database system. Use the following table to find your best starting point.

If you are a(n)	Follow these steps
Database administrator, database architect, database developer, database engineer, or data analyst	**1.** Install the practice files as described in the next section, "Installing and Using the Practice Files." **2.** Work through the chapters in Parts One through Four sequentially. **3.** Complete Parts Five through Seven along with the three online articles as your level of experience and interest dictates.
Application developer	**1.** Install the practice files as described in the next section, "Installing and Using the Practice Files." **2.** Skim the chapters in Part One to get an overview of installing SQL Server and the tools available, and then concentrate on the chapters in Parts Two through Four. **3.** Complete Parts Five through Seven along with the three online articles as your level of experience and interest dictates.
System administrator, network administrator, or security administrator	**1.** Install the practice files as described in the next section, "Installing and Using the Practice Files." **2.** Work through the chapters in Part One. **3.** Skim the chapters in Parts Two through Four. **4.** Work through the chapters in Part Five. **5.** Complete Parts Six and Seven along with the three online articles as your level of experience and interest dictates.
Business analyst	**1.** Install the practice files as described in the next section, "Installing and Using the Practice Files." **2.** Skim through the chapters in Parts One through Six. **3.** Work through the chapters in Part Seven as your level of experience and interest dictates. **4.** Complete the three online articles as your level of experience and interest dictates.
IT Management	**1.** Work through Chapter 1. **2.** Skim through the rest of the chapters and online articles as your experience and level of interest dictates.
Referencing the book after working through the exercises	**1.** Use the Index or the Table of Contents to find information about particular subjects. **2.** Read the Quick Reference sections at the end of each chapter to find a brief review of the syntax and techniques presented in the chapter.

Conventions and Features in This Book

This book presents information using conventions designed to make the information readable and easy to follow. Before you start, read the following list, which explains conventions you'll see throughout the book and points out helpful features that you might want to use.

Conventions

- Each exercise is a series of tasks. Each task is presented as a series of numbered steps (1, 2, and so on). A round bullet (•) indicates an exercise that has only one step.
- Notes labeled "Tip" provide additional information or alternative methods for completing a step successfully.
- Notes labeled "Important" alert you to information you need to check before continuing.
- Text that you type appears in bold.
- A plus sign (+) between two key names means that you must press those keys at the same time. For example, "Press Alt+Tab" means that you hold down the Alt key while you press the Tab key.

Other Features

- Sidebars throughout the book provide more in-depth information about the exercise. The sidebars might contain background information, design tips, or features related to the information being discussed.
- Each chapter ends with a Quick Reference section. The Quick Reference section contains quick reminders of how to perform the tasks you learned in the chapter.

System Requirements

You'll need the following hardware and software to complete the practice exercises in this book:

- Microsoft Windows Vista Home Basic Edition or higher, Windows Server 2008 Standard edition or higher, Windows Server 2003 SP2 or higher, or Window XP Professional SP2 or higher.

Note SQL Server 2008 is not supported on Windows Server 2008 Server Core edition.

- Microsoft SQL Server 2008 Evaluation edition, SQL Server 2008 Developer edition, or SQL Server 208 Enterprise edition.

Note You can use other editions of SQL Server 2008, however, you will be limited by the feature set supported by the SQL Server edition that you have installed.

- 2.0 GHz Pentium III+ processor, or faster
- 1 GB of available, physical RAM
- 2GB of available disk space
- Video (800 × 600 or higher resolution) monitor with at least 256 colors
- CD-ROM or DVD-ROM drive
- Microsoft mouse or compatible pointing device

You will also need to have Administrator access to your computer to configure SQL Server 2008.

Sample Databases

All of the examples within this book utilize the AdventureWorks and AdventureWorksDW sample databases. Sample databases no longer ship with SQL Server and must be downloaded from the CodePlex website at *http://www.codeplex.com/SQLServerSamples*.

Tip In addition to the sample databases, the CodePlex site contains dozens of examples, sample applications, and add-ons that can greatly enhance your SQL Server experience.

Code Samples

The companion CD inside this book contains the code samples that you'll use as you perform the exercises. By using the code samples, you won't waste time creating files that aren't relevant to the exercise. The files and the step-by-step instructions in the lessons also let you learn by doing, which is an easy and effective way to acquire and remember new skills.

Digital Content for Digital Book Readers: If you bought a digital-only edition of this book, you can enjoy select content from the print edition's companion CD.
Visit **http://www.microsoftpressstore.com/title/9780735626041** to get your downloadable content. This content is always up-to-date and available to all readers.

Installing the Code Samples

Follow these steps to install the code samples and required software on your computer so that you can use them with the exercises.

1. Remove the companion CD from the package inside this book and insert it into your CD-ROM drive.

> **Note** An end-user license agreement should open automatically. If this agreement does not appear, open My Computer on the desktop or Start menu, double-click the icon for your CD-ROM drive, and then double-click StartCD.exe.

2. Review the end-user license agreement. If you accept the terms, select the accept option and then click Next.

   A menu will appear with options related to the book.

3. Click Install Code Samples.
4. Follow the instructions that appear.

The code samples are installed to the following location on your computer:

Documents\Microsoft Press\SQL Server 2008 Step By Step.

Using the Code Samples

Each chapter in this book explains when and how to use any code samples for that chapter. When it's time to use a code sample, the book will list the instructions for how to open the files.

Uninstalling the Code Samples

Follow these steps to remove the code samples from your computer.

1. In Control Panel, open Add Or Remove Programs if running Windows XP or Programs\ Uninstall A Program if running Windows Vista.
2. From the list of Currently Installed Programs in Windows XP, or from the list of Uninstall Or Change A Program in Windows Vista, select <Microsoft SQL Server 2008 Step by Step>.
3. Click Remove in Windows XP or click Uninstall/Change in Windows Vista.
4. Follow the instructions that appear to remove the code samples.

Find Additional Content Online

As new or updated material becomes available that complements your book, it will be posted online on the Microsoft Press Online Developer Tools Web site. The type of material you might find includes updates to book content, articles, links to companion content, errata, sample chapters, and more. This Web site is available at *www.microsoft.com/learning/books/online/serverclient*, and is updated periodically.

Support for This Book

Every effort has been made to ensure the accuracy of this book and the contents of the companion CD. As corrections or changes are collected, they will be added to a Microsoft Knowledge Base article.

Microsoft Press provides support for books and companion CDs at the following Web site:

http://www.microsoft.com/learning/support/books/default.mspx.

Questions and Comments

If you have comments, questions, or ideas regarding the book or the companion CD, or questions that are not answered by visiting the sites above, please send them to Microsoft Press via e-mail to:

mspinput@microsoft.com.

Or via postal mail to:

Microsoft Press
Attn: *Programming Microsoft SQL Server 2008 Step by Step* Series Editor
One Microsoft Way
Redmond, WA 98052-6399.

Please note that Microsoft software product support is not offered through the above addresses.

Chapter 3

Using the Tools in SQL Server 2008

After completing this chapter, you will be able to

- Select the appropriate SQL Server 2008 tool for a given task
- Manage SQL Server 2008 services
- Launch, navigate, and utilize SQL Server Management Studio
- Utilize several shortcuts for better productivity
- Configure Database Mail

SQL Server 2008 ships with eight stand-alone tools used to configure, manage, and monitor SQL Server services. Within this group of eight core tools, you can also design SQL Server objects and execute code. The most wide-reaching tool, SQL Server Management Studio (SSMS), contains four additional tools designed for management and monitoring.

Note You can install instances of SQL Server Integration Services, Reporting Services, Analysis Services, and the Database Engine. To simplify the terminology, we will simply refer to a Database Engine instance as a SQL Server instance. All other instances will be referred to as either an SSIS, SSAS, or SSRS instance.

SQL Server Documentation

SQL Server 2008 ships with a very comprehensive Books Online. While many do not consider documentation as a "tool," the saying "information is power" immediately comes to mind. There is a reason that you will hear someone say: "Read the manual." Undeservedly, Books Online has received a very bad reputation.

Books Online should be your primary source for information concerning SQL Server 2008, after this book of course. Books Online contains detailed explanations of every feature within SQL Server, syntax on every command, and thousands of code samples that you can apply. Additionally, SQL Server Books Online integrates a vast array of online content into the local documentation in order to provide extensive, constantly updated information that can be applied within your environment. While we will not spend a significant amount of time going through Books Online, we will explain a couple of very useful and often-overlooked features.

Hidden within Books Online, underneath the How Do I Link button, lurks a set of several dozen comprehensive tutorials that walk you through important feature sets such as policy-based management, *hierarchyID* data types, designing OLAP cubes, implementing Data mining models, deploying replication, and building SSIS packages or SSRS reports, as shown in Figure 3-1.

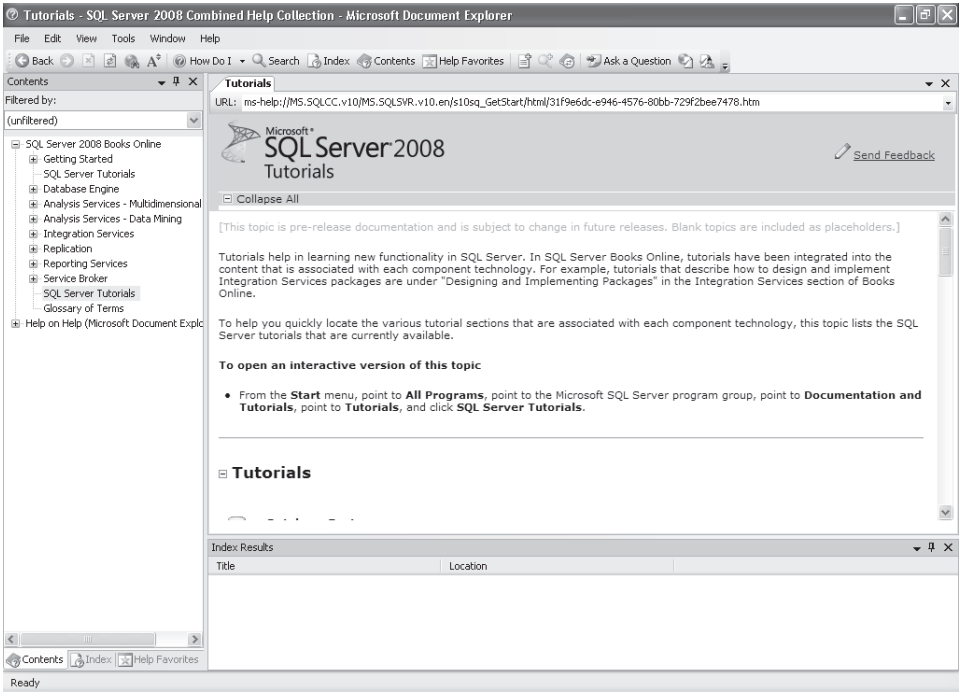


FIGURE 3-1 SQL Server 2008 tutorials

You can also see a set of three links at the far right of the toolbar in Figure 3-1. The first of these three links, Ask A Question, will launch a browser window into the right-hand pane and take you to the Microsoft Developer Network (MSDN) forums. The MSDN forums allow you to ask any question about SQL Server for which you cannot find an answer. Questions are answered by volunteers who include thousands of SQL Server professionals around the world as well as hundreds of members of the SQL Server development team.

The second-to-last link will again take you to the MSDN forums, but will automatically apply a filter to display only those questions you are participating in so that you can easily follow up on the status of your questions.

The last link will launch a browser window in the right-hand pane and take you to Microsoft Connect where you can post feedback on SQL Server or a particular feature

as well as publish bug reports. The bug reports posted are continuously reviewed by the SQL Server development team so that they can proactively supply patches to functionality. The product feedback plays an important role during the planning of the next version of SQL Server.

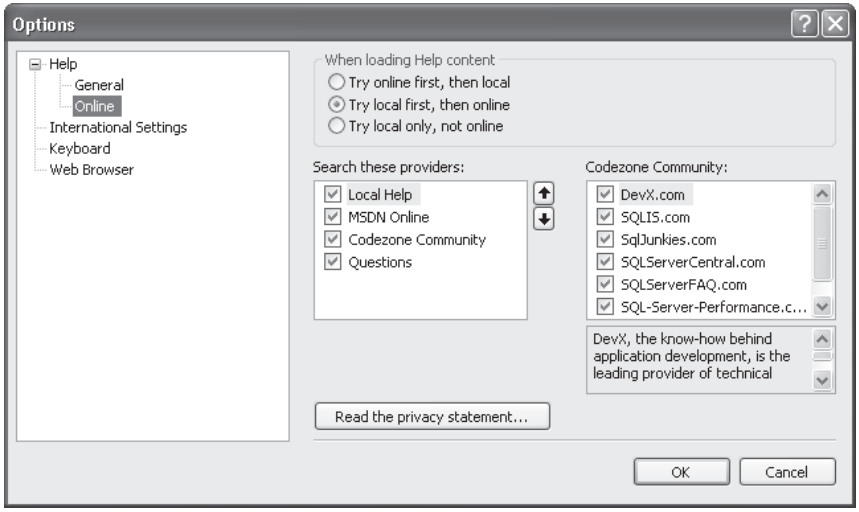
You can bookmark topics that you reference frequently by using the Help Favorites feature. The Help Favorites feature allows you to also save frequently executed searches for later recall.

Note Because SQL Server Books Online contains features to integrate the local version of help with various Web sites, you can experience some performance issues. To mitigate any performance issues, you should set Books Online to look at the local help first before trying online resources.

In this procedure, you will configure Books Online for optimal performance.

Configure Books Online

1. Launch Books Online by selecting Start | All Programs | Microsoft SQL Server 2008 | Documentation And Tutorials | SQL Server Books Online.
2. Select Tools | Options.
3. Select Online and configure as shown here.



Management Tools in SQL Server 2008

SQL Server 2008 ships with a set of nine tools for managing SQL Server instances and interacting with data: OSQL, SQLCMD, Tablediff, Bulk Copy Program (BCP), SQLDiag, Resource Governor, SQL Server Configuration Manager, SSMS, and Database Mail.

OSQL is a command line utility which was added to SQL Server 2000 as a replacement for ISQL. OSQL allows you to connect to and execute queries against a SQL Server instance without requiring the overhead of a graphical interface.

Important OSQL has been deprecated as of SQL Server 2005. While OSQL is still available in SQL Server 2008, you should rewrite any OSQL routines to utilize SQLCMD.

SQL Server 2005 introduced SQLCMD as the command line query interface that replaced OSQL. While OSQL allowed you to submit interactive queries from a command line along with very limited automation capabilities, SQLCMD provides a rich automation interface complete with variable substitution and dynamic code creation/execution.

Note A discussion of the specifics of SQLCMD is beyond the scope of this book. For details on SQLCMD, please see the Books Online topic "SQLCMD Utility."

You can use Tablediff.exe to compare the data between two tables. Tablediff can be run to alert you if the data or structure of two tables is different. Additionally, Tablediff can generate a script file containing the statements necessary to bring the destination table into synchronization with the source table. Tablediff is primarily used within replication architectures.

The BCP utility is the oldest utility within the SQL Server product, dating all the way back to the very first version of SQL Server. BCP has been enhanced with each successive version to handle new data types and named instances, but the speed and feature set has not changed. BCP is used to export data from a table to a file as well as import data from a file into a table. If your import and export needs are reasonably simple and straightforward, BCP should be the only utility that you would need. For more advanced import and export capabilities, you should utilize SQL Server Integration Services (SSIS).

Note You will learn about BCP and its import cousin, BULK INSERT, in Chapter 10, "Data Manipulation." You will learn about SSIS in Chapter 24, "Business Intelligence."

SQLDiag is a utility that collects diagnostic information about a SQL Server instance. SQLDiag is designed to capture Windows performance counters, event logs, SQL Server Profiler traces, SQL Server blocking, and SQL Server configuration information. Primarily used as a data

collection engine for Microsoft Customer Service and Support (CSS) to troubleshoot SQL Server issues, the data collected can also be used by a DBA to analyze SQL Server performance and stability issues.

Note The details of SQLDiag are beyond the scope of this book. Please refer to the Books Online article "SQLdiag Utility" for more information.

Resource Governor is a new feature in SQL Server 2005 that is found within SSMS. The purpose of Resource Governor is to allow a DBA to configure rules around resource allocation such as processor or memory that is then applied to specific queries, users, or groups of users. The goal of Resource Governor is to allow high-priority workloads to take priority over other workloads in order to provide the best response based on user expectations.

Note You will learn about Resource Governor in the article, "Performance and Data Capture Tools," which can be found on the Microsoft Press Online Windows Server and Client Web site at www.microsoft.com/learning/books/online/serverclient.

SQL Server Configuration Manager

Shown in Figure 3-2 on page 36, SQL Server Configuration Manager is responsible for managing SQL Server services and protocols. The primary tasks that you will perform with SQL Server Configuration Manager are:

- Start/Stop/Pause/Restart a service
- Change service accounts and passwords
- Manage the startup mode of a service
- Configure service startup parameters

Once you have completed the initial installation and configuration of your SQL Server services, the primary action that you will perform within SQL Server Configuration Manager is to periodically change service account passwords. When changing service account passwords, you no longer have to restart the SQL Server instance for the new credential settings to take effect.

Important Windows Service Control Applet also has entries for SQL Server services and allows you to change service accounts and passwords. You should never change service accounts or service account passwords using the Windows Service Control Applet. SQL Server Configuration Manager needs to be used, because SQL Server Configuration Manager includes the code to regenerate the service master key that is critical to the operation of SQL Server services.

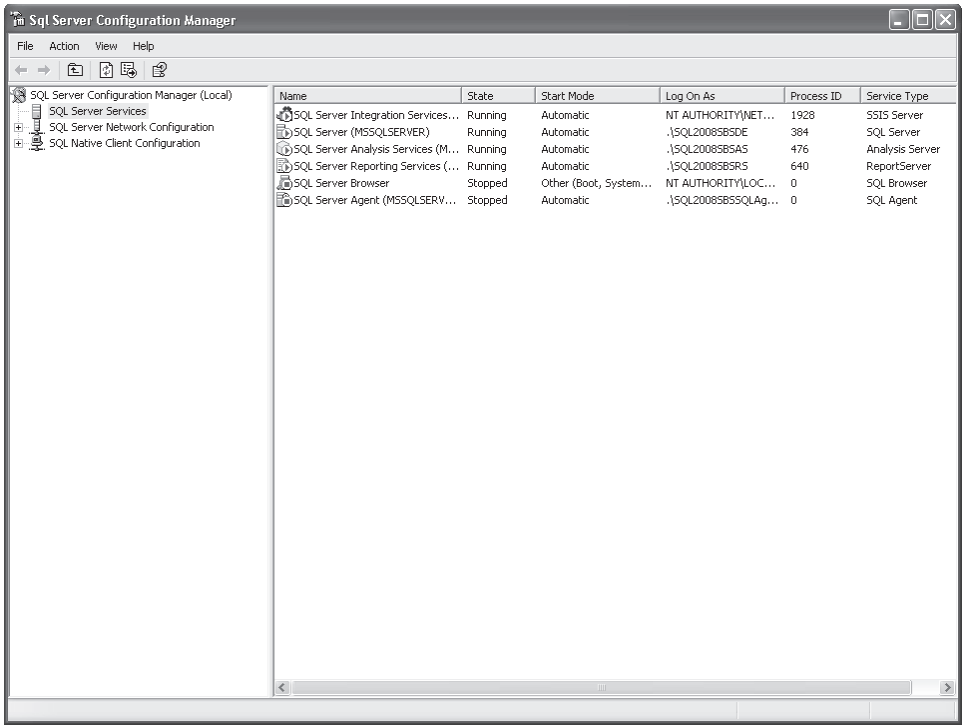


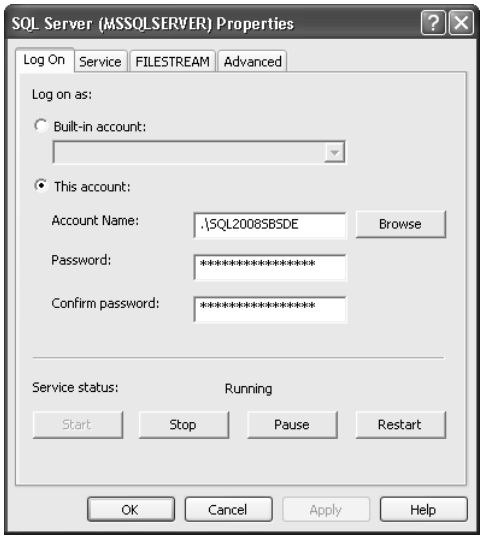
FIGURE 3-2 List of services within SQL Server Configuration Manager

While you can Start, Stop, Pause, and Restart SQL Server services, SQL Server has extensive management features which should ensure that you rarely, if ever, need to shut down or restart a SQL Server service.

In this procedure, you will review the options and settings available for SQL Server services.

Review Service Options

1. Launch SQL Server Configuration Manager by selecting Start | All Programs | Microsoft SQL Server 2008 | Configuration Tools | SQL Server Configuration Manager.
2. In the left-hand pane, highlight SQL Server Services.
3. Double-click the SQL Server service in the right-hand pane to display the Properties dialog box shown here.
4. Review the options on each of the tabs.
5. Click Cancel to close the Properties dialog box without making any changes.



SQL Server Configuration Manager also allows you to configure the communications protocols available to client connections. In addition to configuring protocol-specific arguments, you can also control whether communications are required to be encrypted or whether an instance will respond to an enumeration request, as shown in Figure 3-3.

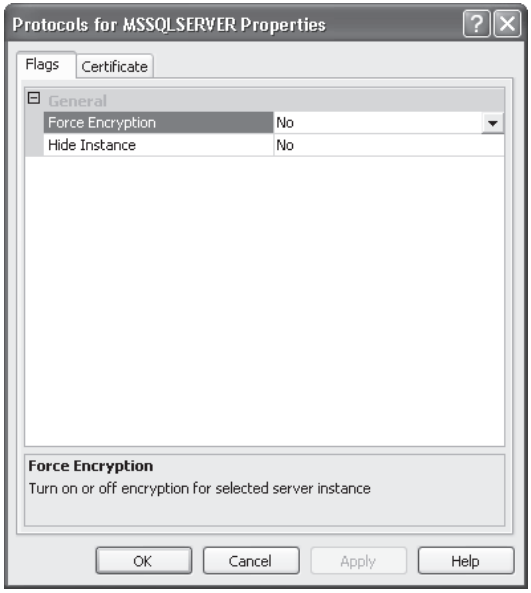


FIGURE 3-3 Protocol properties

Tip Applications can broadcast a special command, called an enumeration request, across a network to locate any SQL Servers that are running on the network. While being able to enumerate SQL Servers is valuable in development and testing environments where instances can appear, disappear, and be rebuilt on a relatively frequent basis, enumeration is not desirable in a production environment. By disabling enumeration responses by setting the Hide Instance to Yes, you prevent someone from using discovery techniques to locate SQL Servers for a possible attack.

SQL Server Management Studio

SQL Server Management Studio is the core tool that you will be spending a large part of your time using. SSMS provides all of the management capabilities for SQL Server services along with the ability to create and execute Transact-SQL (TSQL), Multidimensional Expression (MDX) query language, Data Mining Extensions (DMX), and XML for Analysis (XMLA) code. This section will provide a brief overview of SSMS, as shown in Figure 3-4, to get you started. Each subsequent chapter within this book will extend your knowledge of SSMS capabilities.

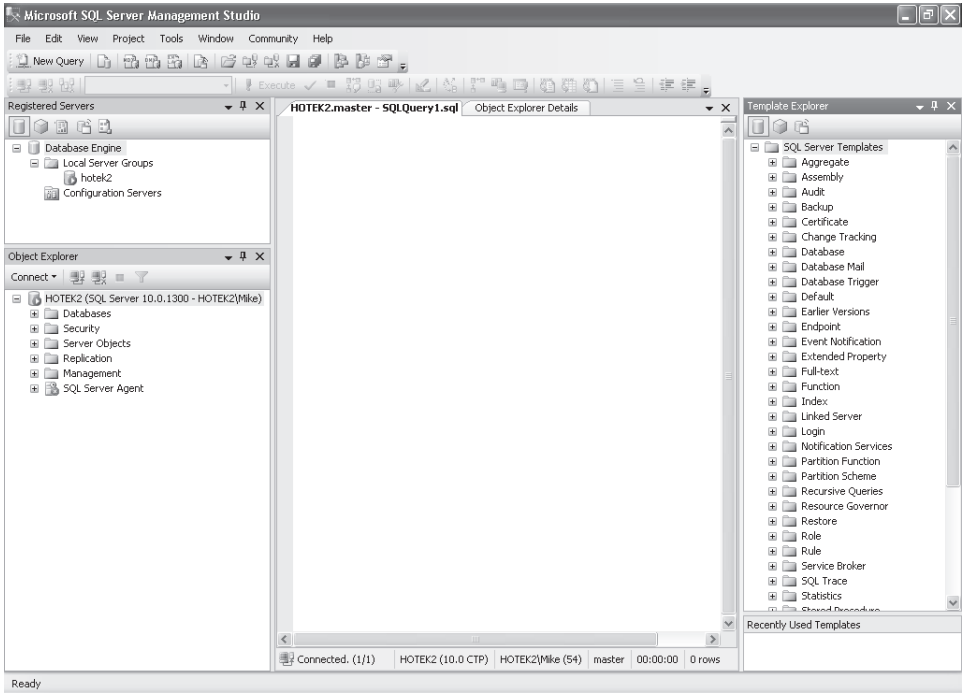


FIGURE 3-4 SQL Server Management Studio

In the following practice, you will launch SSMS and connect to the Database Engine instance that you installed in Chapter 2, "Installing and Configuring SQL Server 2008."

Launch SSMS and Connecting to an Instance

1. Launch SSMS by selecting Start | All Programs | Microsoft SQL Server 2008 | SQL Server Management Studio.
2. When the Connect To Server dialog box is displayed, accept the default options and click Connect.

Note Because you have only installed a default instance at this point, this dialog should default to Database Engine for the server type, <machinename> for the server name, and Windows Authentication for the authentication option. Now that you have connected to an instance within SSMS, for all remaining exercises in this book, we will assume that you can perform these steps and will not repeat them.

SSMS has a variety of windows that you can open and position within the interface in order to access various feature sets.

The Registered Servers window provides a place to store connection information for all of the SQL Server services within your environment. Once stored, you can right-click any server and launch a connection to the server in either the Object Explorer or a query window.

The Template Explorer, shown in the right-hand pane of Figure 3-4, enables access to hundreds of predefined templates to create, alter, or drop objects as well as query various objects using TSQL, MDX, XMLA, or DMX. You can use the templates that ship with SQL Server, modify the templates to include your organization-specific coding standards, and add additional templates or template groups.

The Community menu on the toolbar allows you to launch a browser window into the center pane to access the MSDN forums and Microsoft Connect in the same way as previously described for Books Online.

The Tools | Options menu on the toolbar will display the Options dialog box, as shown in Figure 3-5 on page 40, so that you can set up the SSMS environment specificly the way you want to work.

Configure the SSMS Environment

1. Select Tools | Options from the toolbar.
2. Expand the Environment tree and select the General node. Use the At Startup drop-down list to configure the startup look and feel of SQL Server Management Studio.
3. Expand Text Editor | All Languages | Tabs.
4. Set the Tab Size to 4.

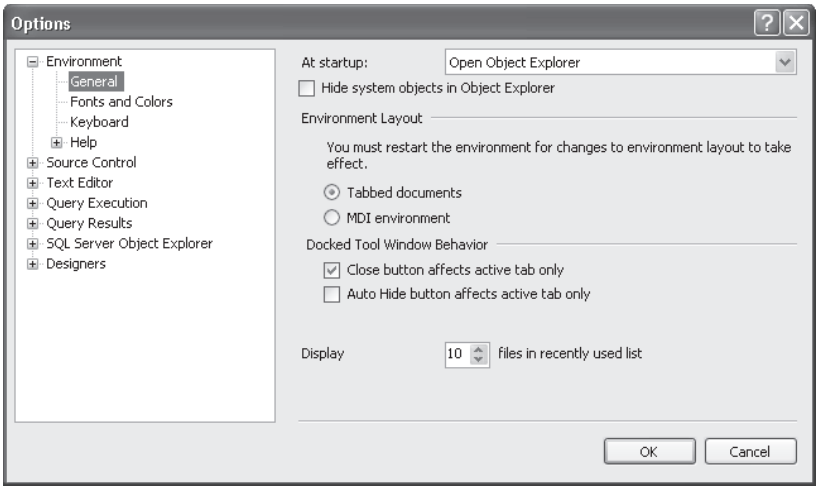


FIGURE 3-5 Options dialog box of the SQL Server Management Studio

5. Set the Indent Size to 4.
6. Select the Insert spaces option.
7. Explore the rest of the options that are available for configuration.
8. Click OK to save your settings.

Tip When you set SSMS to start up with an empty environment, you will not see a Connect To Server dialog and SSMS will immediately start. You will then need to explicitly connect to an instance for the Object Explorer or query window through either the Registered Server pane, File | Connect Object Explorer, or the New Query button. By setting the tab size and insert spaces options, SSMS will automatically replace any tabs with spaces in a query window, allowing you to more easily format and align code even when using a proportional font.

As you can see from Figure 3-6, the Object Explorer provides access to practically any action that you wish to perform against any SQL Server object. You will be using the functionality within the Object Explorer throughout virtually every chapter in this book.

Two additional capabilities of SSMS are object summaries and built-in reporting capabilities. The Object Explorer Details tab will display summary information according to the object that is currently selected within the Object Explorer. SSMS Reports, shown in Figure 3-7, allow you to display either Standard Reports that ship with SQL Server or to access your own custom reports that have been designed using the Reporting Services Report Designer that you will learn about in the article, "Reporting Services," which can be found on the Microsoft Press Online Windows Server and Client Web site at www.microsoft.com/learning/books/online/serverclient.

FIGURE 3-6 Object Explorer for a SQL Server instance

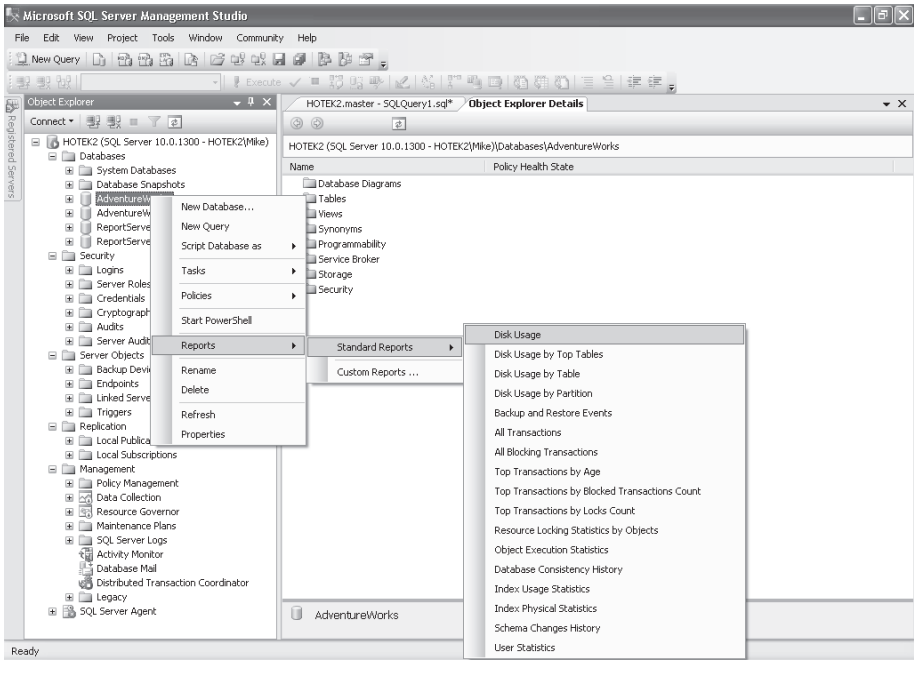


FIGURE 3-7 SSMS Reports

Database Mail

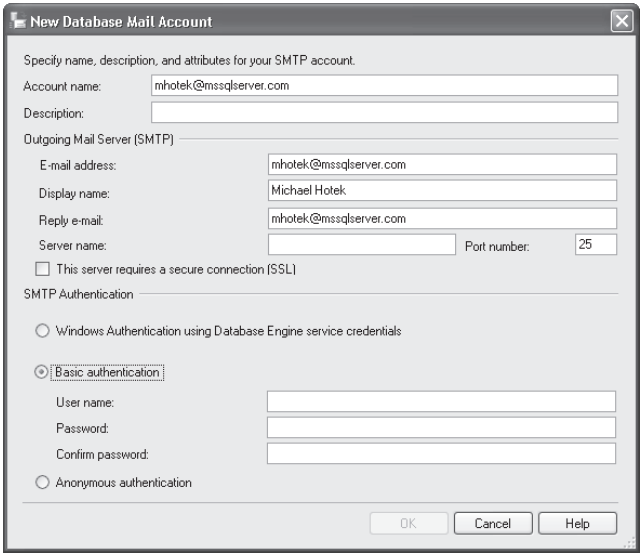
Database Mail enables a SQL Server to send outbound mail messages. While messages can contain the results of queries, Database Mail is primarily used to send alert messages to administrators to notify them of performance conditions or changes that have been made to objects. In the procedure that follows, you will learn how to configure Database Mail.

Configure Database Mail

1. Click the New Query button to open a new query window and execute the following code to enable the Database Mail feature:

```
EXEC sp_configure 'Database Mail XPs',1
GO
RECONFIGURE WITH OVERRIDE
GO
```

2. Within the Object Explorer, open the Management Node, right-click on Database Mail, and select Configure Database Mail.
3. Click Next on the Welcome screen.
4. Select the Set Up Database Mail by Performing the Following Tasks option and click Next.
5. Specify a name for your profile and click Add to specify settings for a mail account.
6. Fill in the account name, e-mail address, display name, reply e-mail, and server name fields on the New Database Mail Account page.
7. Select the appropriate SMTP Authentication mode for your organization and, if using Basic Authentication, specify the user name and password. Your settings should look similar to the following:

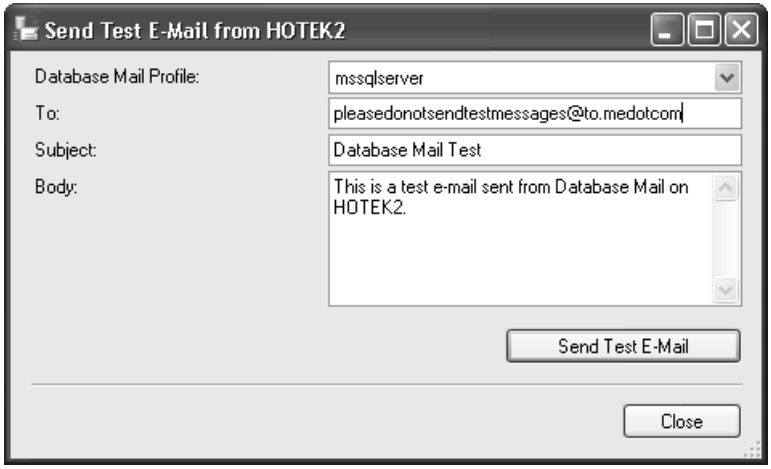


Note Your screen should look similar to the settings in the graphic. I am using my Internet e-mail account and have purposely left the Server Name, User Name, and Password out of the graphic. You will need to fill in the Server Name field if you are using an internal mail server.

8. Click OK and then click Next.
9. Check the box in the Public column next to the profile you just created and set this profile to Yes in the Default Profile column and click Next.
10. Review the settings on the Configure System Parameters page and click Next.
11. Click OK, then click Next, and then click Finish.
12. The final page should show success for all four configuration steps; click Close.
13. Within Object Explorer, right-click SQL Server Agent item and select Start from the shortcut menu in order to start the SQL Server Agent service, if it is not already running.

Note Database Mail utilizes the services of SQL Server Agent to send messages as a background process. If SQL Server Agent is not running, messages will accumulate in a queue within the msdb database.

14. Right-click Database Mail and select Send Test E-mail from the shortcut menu.
15. Select the Database Mail Profile you just created, enter an e-mail address in the To: line, and click Send Test E-Mail.



16. Go to your e-mail client and verify that you have received the test mail message.

Performance Management Tools

In addition to configuration and management tools discussed previously, SQL Server 2008 ships with three specialized tools for capturing, analyzing, and troubleshooting performance data.

Profiler

SQL Server Profiler is a graphical tool that acts as an interface to the SQL Trace Application Programming Interface (API). Profiler allows you to define SQL Server events, as shown in Figure 3-8, that you want to capture information on. You can also specify filtering options to target your data capture within the events that you have specified. You will learn about Profiler in the article, "Performance and Data Capture Tools," which can be found on the Microsoft Press Online Windows Server and Client Web site at www.microsoft.com/learning/books/online/serverclient.

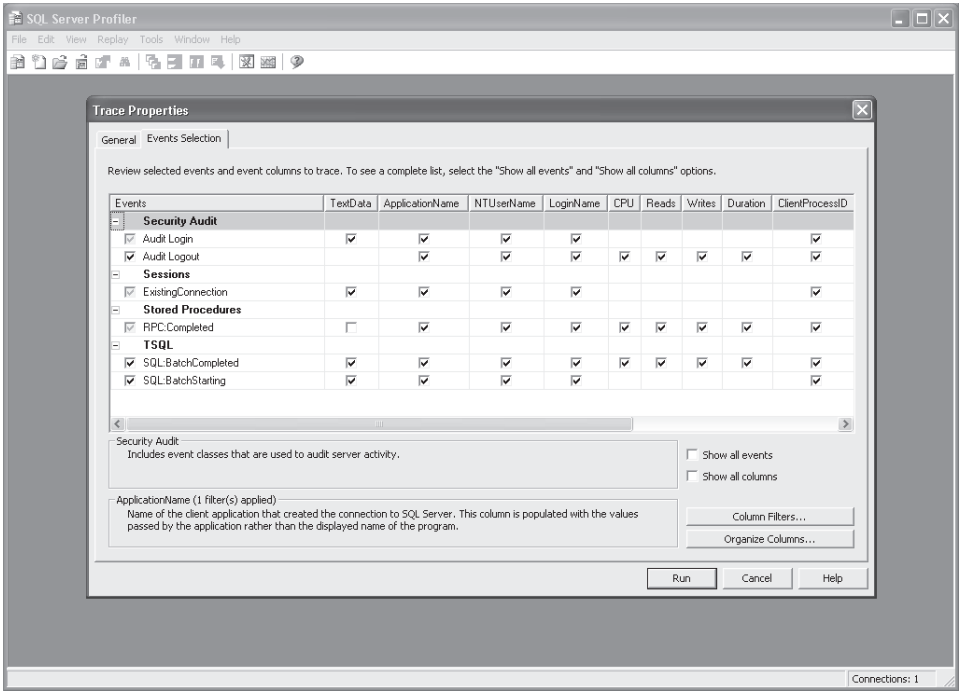


FIGURE 3-8 SQL Server Profiler Trace Properties dialog box Events selection

Database Engine Tuning Advisor

Database Engine Tuning Advisor (DTA) analyzes a query workload and makes recommendations on index and partitioning changes that can improve the performance of your queries (as shown in Figure 3-9). You will learn about indexes in Chapter 6, "Indexes," partitioning in

Chapter 7, "Partitioning," capturing a query workload in the article, "Performance and Data Capture Tools," which can be found on the Microsoft Press Online Windows Server and Client Web site at www.microsoft.com/learning/books/online/serverclient, and how to apply DTA in the article, "Performance Analysis Tools," which can be found on the Microsoft Press Online Windows Server and Client Web site at www.microsoft.com/learning/books/online/serverclient.

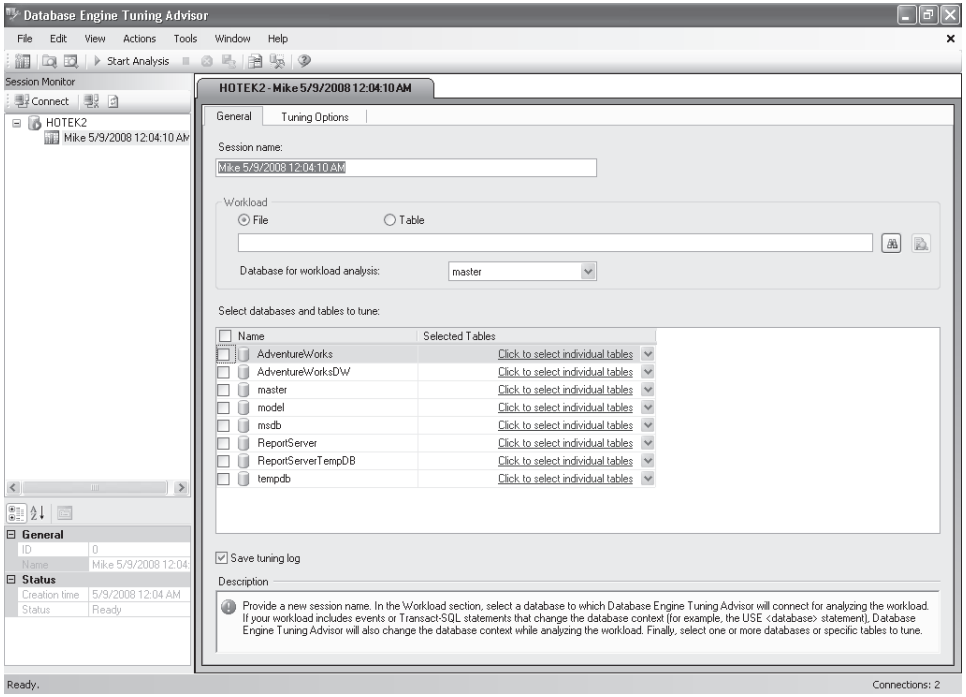


FIGURE 3-9 DTA tuning workload definition

Performance Studio

Performance Studio is the name given to a collection of technologies within SQL Server 2008 that are targeted at the analysis of enterprise-wide performance data. The components of Performance Studio are:

- Performance Data Warehouse
- Data Collectors
- Performance Reports

The Performance Data Warehouse is a database that you create. Data Collectors are SSIS packages, which are executed on a scheduled basis using SQL Server Agent. Performance Reports are a set of Report Designer reports written against the data stored in the Performance Data Warehouse.

You will learn how to configure, manage, and leverage the components of the Performance Studio in the article, "Performance Analysis Tools," which can be found on the Microsoft Press Online Windows Server and Client Web site at www.microsoft.com/learning/books/online/serverclient.

Business Intelligence Tools

Management of SSIS, SSRS, and SSAS occurs within SSMS. However, development of packages, reports, report models, OLAP cubes, and Data mining models occurs within the Business Intelligence Development Studio (BI Dev Studio).

Business Intelligence Development Studio

The BI Dev Studio is the Visual Studio 2008 shell with support for SQL Server 2008 BI projects, as shown in Figure 3-10. Each of the BI projects will be explored in Chapter 24, "Integration Services," Chapter 25, "SQL Server Reporting Services," and Chapter 26, "SQL Server Analysis Services."

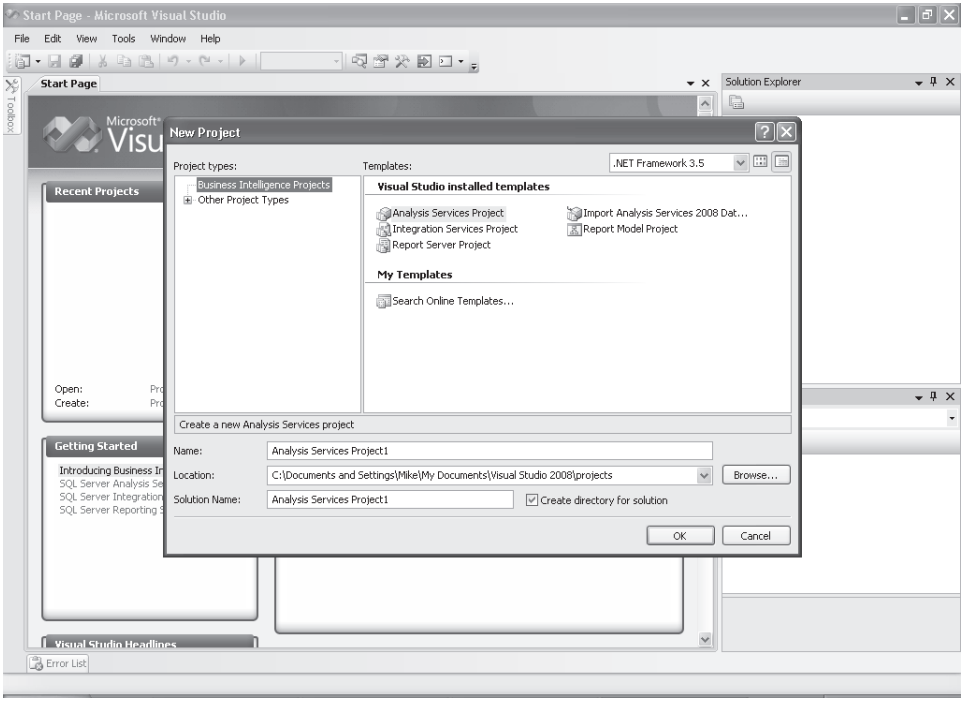


FIGURE 3-10 BI Dev Studio projects

Chapter 3 Quick Reference

To	Do This
Manage a SQL Server instance, OLAP cubes, Data mining models, Integration Services, or Reporting Services	Connect to the appropriate service within the Object Explorer in SQL Server Management Studio
Write and/or execute T-SQL, MDX, or DMX code	Open the appropriate query window (T-SQL, MDX, or DMX) and connect to the instance you want to write or execute code against
Configure and manage Database Mail	Connect to a Database Engine instance
Get help on a topic	Launch SQL Server Books Online. If you are writing a query, you can highlight a term and use SHIFT+F1 to launch Books Online to the highlighted term
Start, Stop, and Pause a service or manage service accounts	Open SQL Server Configuration Manager and either use the Start, Stop, or Pause buttons on the toolbar or double-click the service to access the property sheet
Limit the resources a particular query or user can consume	Configure the Resource Governor within SSMS
Build SSIS, SSRS, or SSAS projects and deploy them to a SQL Server instance	Open BI Dev Studio, create a new project, and design the objects that you wish to deploy
Manage Performance	Use Profiler to capture query workloads, DTA to analyze workloads captured with Profiler, and the Performance Studio to automate the capture of performance metrics

Chapter 14
Triggers

After completing this chapter, you will be able to

- Create DML triggers
- Create DDL triggers

Triggers provide a means to allow you to automatically execute code when an action occurs. Two types of triggers are available in Microsoft SQL Server 2008: DML and DDL. In this lesson, you will learn how to create DML triggers that execute when you add, modify, or remove rows in a table. You will also learn how to create DDL triggers that execute when DDL commands are executed or users log in to an instance.

DML Triggers

Although functions and stored procedures are stand-alone objects, you can't directly execute a trigger. *DML triggers* are created against a table or a view, and are defined for a specific event—*INSERT, UPDATE,* or *DELETE*. When you execute the event a trigger is defined for, SQL Server automatically executes the code within the trigger, also known as "firing" the trigger.

The generic syntax for creating a trigger is:

```
CREATE TRIGGER [ schema_name . ]trigger_name
ON { table | view }
[ WITH <dml_trigger_option> [ ,...n ] ]
{ FOR | AFTER | INSTEAD OF }
{ [ INSERT ] [ , ] [ UPDATE ] [ , ] [ DELETE ] }
[ WITH APPEND ]
[ NOT FOR REPLICATION ]
AS { sql_statement [ ; ] [ ,...n ] | EXTERNAL NAME <method specifier [ ; ] > }
```

When a trigger is defined as AFTER, the trigger fires after the modification has passed all constraints. If a modification fails a constraint check, such as a check, primary key, or foreign key, the trigger is not executed. AFTER triggers are only defined for tables. You can define multiple AFTER triggers for the same action.

A trigger defined with the INSTEAD OF clause causes the trigger code to be executed as a replacement for *INSERT, UPDATE,* or *DELETE*. You can define a single INSTEAD OF trigger for a given action. Although INSTEAD OF triggers can be created against both tables and views, INSTEAD OF triggers are almost always created against views.

Regardless of the number of rows that are affected, a trigger only fires once for an action.

As explained in Chapter 10, "Data Manipulation," SQL Server makes a pair of tables named inserted and deleted available when changes are executed.

In the following exercise, you will create a DML trigger that populates the FinalShipDate column in the Orders.OrderHeader table when the ShipDate column has been populated for all rows in the Orders.OrderDetail table for an OrderID.

Create a DML Trigger

1. Execute the following code against the SQL2008SBS database (the code is from the Chapter14\code1.sql file in the book's accompanying samples):

```
CREATE TRIGGER tiud_orderdetail ON Orders.OrderDetail
FOR INSERT, UPDATE, DELETE
AS

UPDATE a
SET a.FinalShipDate = c.FinalShipDate
FROM Orders.OrderHeader a INNER JOIN
    (SELECT od1.OrderID, MAX(od1.ShipDate) FinalShipDate
    FROM Orders.OrderDetail od1 INNER JOIN
        (SELECT od2.OrderID
        FROM Orders.OrderDetail od2 INNER JOIN inserted i ON od2.OrderID = i.OrderID
        WHERE od2.ShipDate IS NOT NULL
        EXCEPT
        SELECT od3.OrderID
        FROM Orders.OrderDetail od3 INNER JOIN inserted i ON od3.OrderID = i.OrderID
        WHERE od3.ShipDate IS NULL) b
    ON od1.OrderID = b.OrderID
    GROUP BY od1.OrderID) c
ON a.OrderID = c.OrderID
GO
```

2. Validate your newly created trigger by setting the ShipDate column for all order detail rows for an order.

In the following exercise, you will create a DML trigger that enforces referential integrity between the SQL2008SBS and SQL2008SBSFS databases.

Create a DML Trigger

1. Execute the following code against the SQL2008SBS database (the code is from the Chapter14\code2.sql file in the book's accompanying samples):

```
USE SQL2008SBSFS
GO

CREATE TRIGGER tiu_productdocuments ON Products.ProductDocument
FOR INSERT, UPDATE
AS
IF EXISTS (SELECT 1 FROM SQL2008SBS.Products.Product a
            INNER JOIN inserted b ON a.ProductID = b.ProductID)
```

```
BEGIN
    RETURN
END
ELSE
BEGIN
    ROLLBACK TRANSACTION
    RAISERROR('Violation of foreign key',16,1)
END
GO

USE SQL2008SBS
GO

CREATE TRIGGER td_product ON Products.Product
FOR DELETE
AS
IF EXISTS (SELECT 1 FROM SQL2008SBSFS.Products.ProductDocument a
            INNER JOIN deleted b ON a.ProductID = b.ProductID)
BEGIN
    ROLLBACK TRANSACTION
    RAISERROR('You must first delete all documents for this product',16,1)
END
ELSE
BEGIN
    RETURN
END
GO
```

2. Validate your newly created trigger by attempting to insert a document with a ProductID that does not exist.

DDL Triggers

DDL triggers execute under the following circumstances:

- DDL is executed.
- A user logs into an instance.

The general syntax for creating a DDL trigger is as follows:

```
CREATE TRIGGER trigger_name
ON { ALL SERVER | DATABASE }
[ WITH <ddl_trigger_option> [ ,...n ] ]
{ FOR | AFTER } { event_type | event_group } [ ,...n ]
AS { sql_statement [ ; ] [ ,...n ] | EXTERNAL NAME < method specifier > [ ; ] }

<ddl_trigger_option> ::=
  [ ENCRYPTION ]  [ EXECUTE AS Clause ]

<method_specifier> ::=
  assembly_name.class_name.method_name
```

DDL triggers can be scoped at either the database or instance level. To scope a DDL trigger at the instance level, you utilize the ON ALL SERVER option. To scope a DDL trigger at the database level, you utilize the ON DATABASE option.

The following is an example of a DDL trigger:

```
CREATE TRIGGER tddl_tabledropalterprevent
ON DATABASE
FOR DROP_TABLE, ALTER_TABLE
AS
  PRINT 'You are attempting to drop or alter tables in production!'
  ROLLBACK;
```

Note Almost all DDL commands run within the context of a transaction. Since a DDL trigger also runs within the same transaction context, any DDL statement running in the context of a transaction can be rolled back. *ALTER DATABASE* is one of the commands which does not execute in the context of a transaction, because the command affects objects outside of SQL Server that do not obey transactional semantics. Therefore an *ALTER DATBASE* command cannot be rolled back.

The value for the event type is derived from the DDL statement being executed, as listed in Table 14-1.

TABLE 14-1 DDL Trigger Event Types

DDL Command	Event Type
CREATE DATABASE	CREATE_DATABASE
DROP TRIGGER	DROP_TRIGGER
ALTER TABLE	ALTER_TABLE

Event types roll up within a command hierarchy called *event groups*. For example, the CREATE_TABLE, ALTER_TABLE, and DROP_TABLE event types are contained within the DDL_TABLE_EVENTS event group. Event types and event groups allow you to create flexible and compact DDL triggers.

More Info The events and associated event groups that are valid for a DDL triggers can be found in the Books Online article, "Event Groups for Use with DDL Triggers."

Although DML triggers have access to the inserted and deleted tables, DDL triggers have access to the *EVENTDATA()* function which returns the following XML document that can be queried by using the *value()* method available through XQUERY:

```
<EVENT_INSTANCE>
    <EventType>type</EventType>
    <PostTime>date-time</PostTime>
```

```
    <SPID>spid</SPID>
    <ServerName>name</ServerName>
    <LoginName>name</LoginName>
    <UserName>name</UserName>
    <DatabaseName>name</DatabaseName>
    <SchemaName>name</SchemaName>
    <ObjectName>name</ObjectName>
    <ObjectType>type</ObjectType>
    <TSQLCommand>command</TSQLCommand>
</EVENT_INSTANCE>
```

You can retrieve the database, schema, object, and command that you executed, through the following query:

```
SELECT EVENTDATA().value
        ('(/EVENT_INSTANCE/DatabaseName)[1]','nvarchar(max)'),
EVENTDATA().value
        ('(/EVENT_INSTANCE/SchemaName)[1]','nvarchar(max)'),
EVENTDATA().value
        ('(/EVENT_INSTANCE/ObjectName)[1]','nvarchar(max)'),
EVENTDATA().value
        ('(/EVENT_INSTANCE/TSQLCommand)[1]','nvarchar(max)')
```

In the following exercise, you create a DDL trigger to prevent accidentally dropping tables in a production environment.

Create a Database Level DDL Trigger

1. Execute the following code against the SQL2008SBS database (the code is from the Chapter14\code3.sql file in the book's accompanying samples):

   ```
   CREATE TRIGGER tddl_preventdrop
   ON DATABASE
   FOR DROP_TABLE
   AS
       PRINT 'Please disable DDL trigger before dropping tables'
       ROLLBACK TRANSACTION
   GO
   ```

2. Validate your trigger by attempting to drop a table in the SQL2008SBS database.

In the following exercise, you create a logon trigger to limit the number of concurrent connections to a user.

Create an Instance Level DDL Trigger

1. Execute the following code (the code is from the Chapter14\code4.sql file in the book's accompanying samples):

   ```
   CREATE TRIGGER tddl_limitconnections
   ON ALL SERVER
   FOR LOGON
   ```

```
AS
BEGIN
IF (SELECT COUNT(*) FROM sys.dm_exec_sessions
    WHERE is_user_process = 1 AND
        login_name = suser_sname()) > 5

    PRINT 'You are only allowed a maximum of 5 concurrent connections'
    ROLLBACK
END
GO
```

2. Validate your trigger by attempting to create more than five concurrent connections.

Note You have to be careful with a logon trigger, especially one that prevents logging on to the instance. In the exercise above, you had the trigger apply to **all** logins. You should always exclude logins that are members of the sysadmin role, because you do not want to cause a sysadmin to not be able to log in to an instance.

Chapter 14 Quick Reference

To	Do This
Execute code when a DML command is executed	Create a DML trigger
Execute code when a DDL command is executed	Create a DDL trigger

Index

Symbols and Numbers

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X