

Microsoft

MCITP

Exam 70-444



Evaluation
software inside!
SQL Server 2005

15%

EXAM DISCOUNT!

*Limited time offer.
Details inside.*



Optimizing and Maintaining
A Database Administration
Solution by Using
Microsoft

SQL SERVER™ 2005

Self-Paced

Orin Thomas and Ian McLean

Training Kit

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

Copyright © 2006 by Microsoft Corporation and Ian McLean

All rights reserved. No part of the contents of this book may be reproduced or transmitted in any form or by any means without the written permission of the publisher.

ISBN-13: 978-0-7356-2254-8

ISBN-10: 0-7356-2254-X

Library of Congress Control Number 2006932077

Printed and bound in the United States of America.

1 2 3 4 5 6 7 8 9 QWT 0 9 8 7 6 5

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 tkinput@microsoft.com.

Microsoft, Microsoft Press, Active Directory, ActiveX, BizTalk, Excel, Internet Explorer, JScript, MSDN, Visual Basic, Visual C++, Visual C#, Visual J#, Visual SourceSafe, Visual Studio, Windows, Windows NT, and Windows Server are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. Other product and company names mentioned herein may be the trademarks of their respective owners.

The example companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted herein are fictitious. No association with any real company, organization, product, domain name, e-mail address, logo, person, place, or event is intended or should be inferred.

This book expresses the author's views and opinions. The information contained in this book is provided without any express, statutory, or implied warranties. Neither the authors, Microsoft Corporation, nor its resellers, or distributors will be held liable for any damages caused or alleged to be caused either directly or indirectly by this book.

Acquisitions Editor: Ken Jones

Project Editor: Laura Sackerman

Technical Editors: Rozanne Murphy Whalen and Dan Whalen

Indexer: William Meyers

Copy Editor: Roger LeBlanc

Body Part No. X12-48798

For Oksana and Rooslan: I love you.

Orin Thomas

*To my daughter-in-law, Harjit, and my son-in-law, James,
both of whom chose to become related to me of their own free will.*

Ian McLean

Contents at a Glance

1	Troubleshooting Database and Server Performance	1
2	Analyzing Queries	81
3	Failure Diagnosis	155
4	Disaster Recovery	193
5	Performance Monitoring	249
6	Database Maintenance	303
7	SQL Server Integration Services	367
8	Design Data Integrity	419
9	Business Requirements	453
10	Replication	523
11	Security Strategies	583
12	Detecting and Responding to Attacks	647
	Appendix	675



Table of Contents

<i>Acknowledgments</i>	<i>xxiii</i>
Introductionxxv
Hardware Requirements	xxv
Software Requirements	xxvi
Using the CD and DVD	xxvi
How to Install the Practice Tests	xxvi
How to Use the Practice Tests	xxvii
How to Uninstall the Practice Tests	xxviii
Microsoft Certified Professional Program	xxviii
Technical Support	xxix
Evaluation Edition Software Support	xxix
1 Troubleshooting Database and Server Performance1
Before You Begin	1
Lesson 1: Troubleshooting Physical Server Performance	3
Using System Monitor and Performance Logs and Alerts	3
Evaluating Memory Usage	5
Evaluating Disk Usage	8
Evaluating Processor Usage	10
Evaluating Network Usage	11
Evaluating User Connections	12
Solving Resource Problems	13
Generating Performance Counter Logs and Alerts	18
Lesson Summary	25
Lesson Review	25
Lesson 2: Troubleshooting Connectivity to a SQL Server Instance	28
Troubleshooting Tools	28
Analyzing Tempdb Database Issues	33
Monitoring Instance Memory Usage	35
Data Caching	35
Analyzing Statement Recompiles	36

What do you think of this book?
We want to hear from you!

Microsoft is interested in hearing your feedback about this publication so we can continually improve our books and learning resources for you. To participate in a brief online survey, please visit: www.microsoft.com/learning/booksurvey/

Configuring Connections.....	38
SQL CLR Memory Usage	40
Configuring CPU Parallelism	41
Monitoring Waits and Queues	43
Using SQL Server Profiler.....	46
Lesson Summary.....	50
Lesson Review	51
Lesson 3: Troubleshooting Database Performance	53
Resolving Space Issues	53
Monitoring Auto-Grow and Auto-Shrink	55
Updating Statistics	55
Evaluating Index Usage	57
Auditing and Analyzing Poorly Written Queries	61
Monitoring Transaction Log Size	63
Monitoring Database Growth.....	64
Investigating Locks and Deadlocks	67
Optimizing RAID Configuration	69
Troubleshooting Database and Transaction Log Storage	70
Using the Database Engine Tuning Advisor	70
Lesson Summary.....	74
Lesson Review	74
Chapter Review.....	76
Chapter Summary.....	76
Key Terms.....	77
Case Scenario	77
Case Scenario: Resolving Physical Server and Database Bottlenecks.....	78
Suggested Practices	79
Troubleshoot Physical Server Performance.....	79
Troubleshoot Instance Performance	79
Troubleshoot Database Performance.....	80
Take a Practice Test.....	80
2 Analyzing Queries.....	81
Before You Begin	82
Lesson 1: Identifying Poorly Performing Queries	83
Using Query Editor.....	83
Using SQL Server Profiler.....	86

Using the Database Tuning Advisor	87
Using SQL Trace	88
Using DMVs	89
Identifying a Badly Performing Query	93
Lesson Summary	96
Lesson Review	97
Lesson 2: Analyzing a Query Plan to Detect Inefficiencies in Query Logic	98
Detecting Excessive I/O Activity	98
Monitoring Table Scans	102
Monitoring CPU Utilization	107
Obtaining Query Plan Statistics	108
Lesson Summary	110
Lesson Review	110
Lesson 3: Maintaining and Optimizing Indexes	112
Defragmenting an Index	112
Reorganizing and Rebuilding an Index	113
Adding an Index	116
Specifying the Fill Factor	117
Using the PAD_INDEX Option	118
Using Clustered and Nonclustered Indexes	118
Using Covering Indexes	121
Using Indexed Views	123
Creating XML Indexes	124
Creating Partitioned Indexes	126
Performing Index Analysis	127
Lesson Summary	130
Lesson Review	130
Lesson 4: Enforcing Appropriate Stored Procedure Logging and Output	132
Handling Exceptions	134
Examining the Default Log Trace File	134
Lesson Summary	135
Lesson Review	136
Lesson 5: Troubleshooting Concurrency Issues	137
Using the SQL Server Locks Performance Counters	137
Evaluating the Transactions/sec Performance Counter	142
Using Alerts to Trigger the Notification Process	144
Using SQL Server Profiler to Troubleshoot Concurrency Issues	146

Saving a Deadlock Graph	146
Lesson Summary	148
Lesson Review	149
Chapter Review	151
Chapter Summary	151
Key Terms	151
Case Scenario	152
Case Scenario: Dealing with Compatibility Problems and Fragmented Indexes	152
Suggested Practices	153
Identify Poorly Performing Queries	153
Analyze a Query Plan to Detect Inefficiencies in Query Logic	153
Maintain and Optimize Indexes	154
Enforce Appropriate Stored Procedure Logging and Output	154
Troubleshoot Concurrency Issues	154
Take a Practice Test	154
3 Failure Diagnosis	155
Before You Begin	155
Lesson 1: Diagnosing Database Failures	157
Log File Viewer	157
Filtering Logs	158
Understanding Database Engine Errors	159
Diagnosing Common Problems Using Logs	162
Filtering a Log Using the Log File Viewer	165
Lesson Summary	166
Lesson Review	167
Lesson 2: Diagnosing Physical Server Failures	169
Diagnosing Volumes and Disks	170
Diagnosing RAM and Processor Problems	174
Diagnosing Other Hardware Problems	175
Using CHKDSK	176
Lesson Summary	176
Lesson Review	176
Lesson 3: SQL Server Service Failures	178
SQL Server 2005 Services	178
Service Password Expiration	184

SQL Browser Service and DAC	185
The SQL Server Agent Service	186
Configuring a Service to Automatically Restart	186
Lesson Summary	187
Lesson Review	187
Chapter Review	189
Chapter Summary	189
Key Terms	190
Case Scenarios	190
Case Scenario 1: Diagnosing Database Configuration Errors	190
Case Scenario 2: Diagnosing Database Hardware Errors	191
Suggested Practices	191
Diagnose Causes of Failures	191
Take a Practice Test	192
4 Disaster Recovery	193
Before You Begin	194
Lesson 1: Planning for Fault Tolerance	195
SQL Server and RAID	195
Failover Clustering	197
Database Mirroring	198
Log Shipping	200
Configure Log Shipping	202
Lesson Summary	205
Lesson Review	205
Lesson 2: Recovering from Failure	207
Restoring the System Databases	207
Recovery Models	209
Files and Filegroups	212
Backup Types	215
Snapshot Backups	217
Use the Full Recovery Model and Back Up a Database	218
Lesson Summary	220
Lesson Review	220
Lesson 3: Recovering from Database Disaster	222
Restoration and Rolling Forward	222
Restore Types	224

Safety Features	228
Database Snapshots	229
Troubleshooting Orphaned Users	232
Create and Revert to a Snapshot	233
Lesson Summary	234
Lesson Review	234
Lesson 4: Salvaging Data from a Damaged Database	237
Restoring Data from Bad Tapes	237
Using DBCC CHECKDB to Repair Data.	238
Rebuilding Indexes	239
Managing Suspect Pages.	240
Check a Database for Errors	241
Lesson Summary	242
Lesson Review	242
Chapter Summary.	244
Key Terms.	244
Case Scenarios.	245
Case Scenario 1: Ensuring Fault Tolerance	245
Case Scenario 2: Backup and Recovery.	246
Suggested Practices	246
Plan for Fault Tolerance	246
Salvage Good Data from a Damaged Database by Using Restoration Techniques	247
Recover from a Database Disaster	247
Recover from a Failure of SQL Server 2005.	247
Take a Practice Test.	247
5 Performance Monitoring	249
Before You Begin	250
Lesson 1: Defining and Implementing Monitoring Standards for a Physical Server	252
Establishing Performance Thresholds	252
Establishing Performance Baselines	254
Deciding What to Monitor	256
Using Performance Counters	257
Defining Traces	263
Setting Alerts.	266
Setting Event Notifications	267

Setting an Alert	269
Lesson Summary	273
Lesson Review	274
Lesson 2: Choosing the Appropriate Information to Monitor	277
Using the <i>sys.dm_exec_query_stats</i> DMV	277
Using the SQL Server Log	281
Analyzing Waits	282
Tracing Resource Usage	284
Checking Service Availability and Status	288
Viewing and Recycling the SQL Error Log	291
Lesson Summary	295
Lesson Review	296
Chapter Review	298
Chapter Summary	298
Key Terms	299
Case Scenarios	299
Case Scenario 1: Automating, Monitoring, and Configuring Alerts	299
Case Scenario 2: Identifying Slow and Resource-Intensive Transactions	300
Suggested Practices	300
Define and Implement Monitoring Standards for a Physical Server	301
Choose the Information to Monitor	301
Take a Practice Test	302
6 Database Maintenance	303
Before You Begin	304
Lesson 1: Creating and Implementing a Maintenance Strategy for Database Servers	305
Capturing Data Definition Language (DDL) Operations Using DDL Triggers	305
Creating Database Diagrams	307
Job Dependency Diagrams	308
Applying Service Packs, Software Updates, and Security Updates	310
Creating a Database Diagram for the AdventureWorks DW Database	314
Lesson Summary	314
Lesson Review	315
Lesson 2: Designing a Database Maintenance Plan	317
Database Maintenance Plans	317
Database Backups	324

Manual Maintenance	327
Create a Backup Device and Take a Full Backup	328
Lesson Summary	328
Lesson Review	329
Lesson 3: Managing Reporting Services	331
The Report Server Database	331
Report Manager	332
Creating a Basic Report	333
Report Snapshots	337
Report Subscriptions	339
Reporting Services Configuration Manager	340
Configuring Role-Based Security	342
Moving a Report Server	346
Generate a Report	348
Lesson Summary	349
Lesson Review	349
Lesson 4: Designing a Strategy to Manage Data Across Linked Servers	351
Linked Server Basics	351
How Linked Servers Work	351
Setting Up Linked Servers	352
Configuring OLE DB Provider Options	355
Configuring Linked Servers for Delegation	356
Linked Server Security	358
Configuring Linked Server Options	359
Lesson Summary	360
Lesson Review	360
Chapter Review	362
Chapter Summary	362
Key Terms	363
Case Scenarios	363
Case Scenario 1: Managing Updates	363
Case Scenario 2: Configuring Report Server Roles	364
Suggested Practices	364
Take a Practice Test	365
7 SQL Server Integration Services	367
Before You Begin	368

Lesson 1: Constructing SSIS Packages	369
Business Intelligence Development Studio	369
SSIS Packages	372
Building a Package	374
Executing Packages	377
Creating an SSIS Package	378
Lesson Summary	384
Lesson Review	384
Lesson 2: Securing SSIS Packages	386
Securing Sensitive Data Using Package Protection Levels	386
Database-Level Role Security	387
Package and Configuration Storage Security	390
Digitally Signing Packages	390
Securing an SSIS Package	391
Lesson Summary	392
Lesson Review	393
Lesson 3: Troubleshooting SSIS Packages	395
Package Checkpoints	395
Incorporating Transactions into Packages	397
Package Debugging	398
Data Viewers	398
Breakpoints Window	399
Package Logging	400
Setting Checkpoints	402
Lesson Summary	403
Lesson Review	403
Lesson 4: Deploying SSIS Packages	405
Package Configurations	405
Deployment Utilities	408
Deployment of Packages	409
Scheduling Package Execution	412
Schedule Package Execution	412
Lesson Summary	413
Lesson Review	414
Chapter Review	415
Chapter Summary	415
Key Terms	416

Case Scenarios	416
Case Scenario 1: Creating and Managing SSIS Packages	416
Case Scenario 2: SSIS Package Administration	417
Suggested Practices	418
Design and Manage SQL Server Integration Services (SSIS) Packages	418
Take a Practice Test	418
8 Design Data Integrity	419
Before You Begin	420
Lesson 1: Reconciling Data Conflicts	421
Detecting Conflicts	421
Resolving Conflicts	422
Viewing Data Conflicts	425
Viewing Conflicts	426
Lesson Summary	426
Lesson Review	427
Lesson 2: Making Implicit Constraints Explicit	428
Understanding Implicit and Explicit Constraints	428
Constraints	429
Triggers	438
Configuring a Check Constraint	439
Lesson Summary	440
Lesson Review	441
Lesson 3: Assigning Data Types to Control Characteristics of Data Stored in a Column	442
Transact-SQL Data Types	442
Alias Data Types	444
User-Defined CLR Types	446
Creating an Alias Data Type	447
Lesson Summary	448
Lesson Review	448
Chapter Review	449
Chapter Summary	449
Key Terms	450
Case Scenarios	450
Case Scenario 1: Making Implicit Constraints Explicit	450
Case Scenario 2: Data Types	451

Suggested Practices	451
Design Data Integrity	451
Take a Practice Test	452
9 Business Requirements	453
Before You Begin	454
Lesson 1: Enforcing Data Quality According to Business Requirements	455
Analyzing Business and Regulatory Requirements and Determining Exceptions	456
Establishing Business Requirements for Data Quality	457
Ensuring Applications Enforce Data Quality	460
Using SQL Server Integration Services	465
Using Fuzzy Transformations	468
Using Data Mining	472
Creating Queries to Inspect the Data	475
Using CHECKSUM	482
Cleaning Data	483
Using the UNION Operator	487
Lesson Summary	491
Lesson Review	492
Lesson 2: Optimizing a Database Change Control Strategy to Meet Business Requirements	494
Verifying that Database Change Control Procedures Are Being Followed	494
Identifying All Database Objects Related to a Particular Deployment	501
Using DDL Triggers	509
Lesson Summary	515
Lesson Review	516
Chapter Review	518
Chapter Summary	518
Key Terms	518
Case Scenarios	519
Case Scenario 1: Checking and Correcting Invalid Database Entries	519
Case Scenario 2: Managing Schema Changes	520
Suggested Practices	521
Enforce Data Quality According to Business Requirements	521
Optimize a Database Change Control Strategy to Meet Business Requirements	521
Take a Practice Test	522

10	Replication	523
	Before You Begin	523
	Lesson 1: Designing a Strategy to Manage Replication	525
	Selecting a Replication Strategy	526
	Specifying a Replication Type, Topology, and Model	527
	Designing and Configuring Replication Alerts	534
	Monitoring Replication Status	537
	Verifying Replication	543
	Resolving Replication Conflicts	548
	Configuring Agent Profiles	554
	Tuning Replication Configuration	556
	Using Replication with Database Mirroring	561
	Configuring and Verifying Replication	565
	Lesson Summary	575
	Lesson Review	575
	Chapter Review	579
	Chapter Summary	579
	Key Terms	579
	Case Scenarios	580
	Case Scenario 1: Selecting Replication Type and Model	580
	Case Scenario 2: Tuning Replication	581
	Suggested Practices	582
	Design a Strategy to Manage Replication	582
	Take a Practice Test	582
11	Security Strategies	583
	Before You Begin	584
	Lesson 1: Maintaining a Server-Level Security Strategy	585
	Specifying and Auditing Windows Account Permissions	585
	Auditing SQL Server Service Access	589
	Auditing Server Logins	590
	Assigning the Appropriate Minimum Level of Privileges	593
	Applying the Principle of Least Privilege	594
	Maintaining an Encryption Strategy	595
	Applying Service Packs and Security Updates	600
	Configuring the Surface Area	603
	Using the SQL Server Surface Area Configuration Tool	605

Lesson Summary	609
Lesson Review	609
Lesson 2: Maintaining a User-Level Security Strategy	612
Verifying the Existence and Enforcement of Account Policies	612
Verifying SQL Server Login Authentication.	616
Verifying Permissions on SQL Server Roles and Accounts	623
Using Object Explorer.	633
Lesson Summary	640
Lesson Review	640
Chapter Review	642
Chapter Summary	642
Key Terms	643
Case Scenarios	643
Case Scenario 1: Configuring Security on SQL Server 2005 Member Servers.	643
Case Scenario 2: Adding Your Team Members' User Accounts to Database Roles	644
Suggested Practices	644
Maintain a Server-Level Security Strategy	644
Maintain a User-Level Security Strategy.	645
Take a Practice Test	645
12 Detecting and Responding to Attacks.	647
Before You Begin	648
Lesson 1: Auditing the Existing Infrastructure	649
Analyzing Physical Server Security	649
SQL Server Security Considerations	652
Security Configuration And Analysis	654
Using the MBSA Tool to Audit Security	656
Configuring Security Using Templates	656
Lesson Summary	657
Lesson Review	658
Lesson 2: Protecting Against Threats and Attacks	660
Preparing for and Responding to SQL Server Injection Attacks.	660
Responding to Virus and Worm Attacks	662
Responding to Denial of Service Attacks.	665
Responding to a Denial of Service Attack	666
Responding to Internal Attacks	667

Securing Database Mail	668
Lesson Summary	670
Lesson Review	670
Chapter Review	672
Chapter Summary	672
Key Terms	672
Case Scenarios	673
Case Scenario 1: Physically Securing a Server Room	673
Case Scenario 2: Responding to a Denial of Service Attack	673
Suggested Practices	674
Performing a Security Audit of the Existing Security Infrastructure Based on the Security Plan	674
Prepare for and Respond to Threats and Attacks	674
Take a Practice Test	674
Appendix	675
Configuring the Computers	676
Installing and Configuring the Windows Server 2003 R2 180-Day Evaluation Software	676
Installing SQL Server 2005 Enterprise Edition 180-Day Evaluation	677
Installing SQL Server 2005 Service Pack 1	678
Installing Sample Databases	679
Glossary	681
Answers	691
Index	745

What do you think of this book?
We want to hear from you!

Microsoft is interested in hearing your feedback about this publication so we can continually improve our books and learning resources for you. To participate in a brief online survey, please visit: www.microsoft.com/learning/booksurvey/

Acknowledgments

Producing a book is always a team effort, and the authors would like to thank our program manager, Ken Jones, for commissioning us to write the book and for his assistance throughout the project, and our content development manager, Karen Szall, for guiding us through the early stages. Our main contact was our project editor, Laura Sackerman, and we are most appreciative of her professionalism and patience in getting this difficult project to the finish line.

We would also like to thank our technical editors, Rozanne Murphy Whalen and Dan Whalen, our copy editor, Roger LeBlanc, and our proofreader, Victoria Thulman, all of whom provided valuable and constructive input—particularly to the sections that we wrote at five in the morning.

The work done behind the scenes is always important, and we would like to pay tribute to the professionalism of our indexer, William Meyers, our proofreading coordinator, Sandi Resnick, our layout coordinators, Carl Diltz and Elizabeth Hansford, and our indexing coordinators, Patty Masserman and Shawn Peck.

Few creatures are more antisocial than an author in mid-book. As always, we are truly grateful to our wives, Oksana Thomas and Anne McLean, for their support and infinite patience. Finally we would like to thank the unknown genius who figured out what the coffee bean is for. Without caffeine this book would not have been written.

Introduction

This training kit is designed for experienced database administrators (DBAs) who plan to take Microsoft Certified Information Technology Professional (MCITP) Exam 70-444, as well as for database professionals whose tasks might include defining high-availability solutions, automating administrative procedures, defining security solutions, designing and executing deployments, and monitoring and troubleshooting database servers. We assume that before you begin using this kit you will have a good working knowledge of Microsoft Windows, network technologies, relational databases and their design, Transact-SQL, and the Microsoft SQL Server 2005 client tools.

By using this training kit, you'll learn how to do the following:

- Optimize the performance of database servers and databases.
- Optimize and implement a data recovery plan for a database.
- Design a strategy to monitor and maintain a database solution.
- Design a database data management strategy.
- Design a strategy to manage and maintain database security.

Hardware Requirements

We recommend that you use an isolated network that is not part of your production network to do the practice exercises in this book. You need a two-station network that you can implement either by using two computers connected by a crossover network cable or by using a single computer running virtual machine software. Your computer or computers should meet the following hardware specification:

- Personal computer with a 600-MHz Pentium III-compatible or faster processor (Pentium IV or equivalent if you plan to use virtual machine software)
- 512 MB of RAM (1.5 GB if you plan to use virtual machine software)
- 10 GB of available hard disk space (20 GB if you plan to use virtual machine software)
- DVD-ROM drive
- Super VGA (1024 × 768) or higher resolution video adapter and monitor
- Keyboard and Microsoft mouse, or compatible pointing device

Software Requirements

The following software is required to complete the practice exercises:

- Microsoft Windows 2003 Server with Service Pack 2 (SP2) or later
- Microsoft SQL Server 2005 Enterprise Edition, SP1 or later (A 180-day evaluation edition of Microsoft SQL Server 2005 Enterprise Edition is included on the DVD that comes with this book.)
- The latest version of the AdventureWorks database (which you can find at <http://www.microsoft.com/downloads>)

MORE INFO Software requirements

For more details about these software requirements, please see the Appendix.

Using the CD and DVD

A companion CD and an evaluation software DVD are included with this training kit. The companion CD contains the following:

- **Practice tests** You can reinforce your understanding of how to optimize and maintain a database administration solution by using electronic practice tests you customize to meet your needs from the pool of Lesson Review questions in this book. Or you can practice for the 70-444 certification exam by using tests created from a pool of 300 realistic exam questions, which give you many different practice exams to ensure that you're prepared.
- **An eBook** An electronic version (eBook) of this book is included for times when you don't want to carry the printed book with you. The eBook is in Portable Document Format (PDF), and you can view it by using Adobe Acrobat or Adobe Reader.

The evaluation software DVD contains a 180-day evaluation edition of SQL Server 2005 Enterprise Edition, in case you want to use it with this book.

How to Install the Practice Tests

To install the practice test software from the companion CD to your hard disk, do the following:

1. Insert the companion CD into your CD drive, and accept the license agreement. A CD menu appears.

NOTE If the CD menu doesn't appear

If the CD menu or the license agreement doesn't appear, AutoRun might be disabled on your computer. Refer to the Readme.txt file on the CD-ROM for alternate installation instructions.

2. Click the Practice Tests item, and follow the instructions on the screen.

How to Use the Practice Tests

To start the practice test software, follow these steps:

1. Click Start/All Programs/Microsoft Press Training Kit Exam Prep. A window appears that shows all the Microsoft Press training kit exam prep suites installed on your computer.
2. Double-click the lesson review or practice test you want to use.

NOTE Lesson reviews vs. practice tests

Select the (70-444) Microsoft SQL Server 2005—Optimizing and Maintaining a Database Administration Solution *lesson review* to use the questions from the “Lesson Review” sections of this book. Select the (70-444) Microsoft SQL Server 2005—Optimizing and Maintaining a Database Administration Solution *practice test* to use a pool of 300 questions similar to those in the 70-444 certification exam.

Lesson Review Options

When you start a lesson review, the Custom Mode dialog box appears so that you can configure your test. You can click OK to accept the defaults, or you can customize the number of questions you want, how the practice test software works, which exam objectives you want the questions to relate to, and whether you want your lesson review to be timed. If you're retaking a test, you can select whether you want to see all the questions again or only questions you missed or didn't answer.

After you click OK, your lesson review starts.

- To take the test, answer the questions and use the Next, Previous, and Go To buttons to move from question to question.
- After you answer an individual question, if you want to see which answers are correct—along with an explanation of each correct answer—click Explanation.
- If you'd rather wait until the end of the test to see how you did, answer all the questions and then click Score Test. You'll see a summary of the exam objectives you chose and the percentage of questions you got right overall and per objective. You can print a copy of your test, review your answers, or retake the test.

Practice Test Options

When you start a practice test, you choose whether to take the test in Certification Mode, Study Mode, or Custom Mode:

- **Certification Mode** Closely resembles the experience of taking a certification exam. The test has a set number of questions, it's timed, and you can't pause and restart the timer.
- **Study Mode** Creates an untimed test in which you can review the correct answers and the explanations after you answer each question.
- **Custom Mode** Gives you full control over the test options so that you can customize them as you like.

In all modes, the user interface when you're taking the test is basically the same, but with different options enabled or disabled depending on the mode. The main options are discussed in the previous section, "Lesson Review Options."

When you review your answer to an individual practice test question, a "References" section is provided that lists where in the training kit you can find the information that relates to that question and provides links to other sources of information. After you click Test Results to score your entire practice test, you can click the Learning Plan tab to see a list of references for every objective.

How to Uninstall the Practice Tests

To uninstall the practice test software for a training kit, use the Add Or Remove Programs option in Windows Control Panel.

Microsoft Certified Professional Program

The Microsoft certifications provide the best method to prove your command of current Microsoft products and technologies. The exams and corresponding certifications are developed to validate your mastery of critical competencies as you design and develop, or implement and support, solutions with Microsoft products and technologies. Computer professionals who become Microsoft-certified are recognized as experts and are sought after industry-wide. Certification brings a variety of benefits to the individual and to employers and organizations.

MORE INFO All the Microsoft certifications

For a full list of Microsoft certifications, go to www.microsoft.com/learning/mcp/default.asp.

Technical Support

Every effort has been made to ensure the accuracy of this book and the contents of the companion CD. If you have comments, questions, or ideas regarding this book or the companion CD, please send them to Microsoft Press by using either of the following methods:

E-mail: tkinput@microsoft.com

Postal Mail:

Microsoft Press

Attn: MCTS Self-Paced Training Kit (Exam 70-444): Microsoft SQL Server 2005—Optimizing and Maintaining a Database Administration Solution Editor

One Microsoft Way

Redmond, WA 98052-6399

For additional support information regarding this book and the CD-ROMs (including answers to commonly asked questions about installation and use), visit the Microsoft Press Technical Support Web site at www.microsoft.com/learning/support/books/. To connect directly to the Microsoft Knowledge Base and enter a query, visit <http://support.microsoft.com/search/>. For support information regarding Microsoft software, please go to <http://support.microsoft.com>.

Evaluation Edition Software Support

The 180-day evaluation edition provided with this training kit is not the full retail product and is provided only for the purposes of training and evaluation. Microsoft and Microsoft Technical Support do not support this evaluation edition.

Information about any issues relating to the use of this evaluation edition with this training kit is posted to the Learning Support section of the Microsoft Press Web site (www.microsoft.com/learning/support/books/). For information about ordering the full version of any Microsoft software, please call Microsoft Sales at (800) 426-9400 or visit www.microsoft.com.

Chapter 1

Troubleshooting Database and Server Performance

When a Microsoft SQL Server 2005 server is not meeting its performance requirements, this situation could be a result of issues within a database, issues relating to instances of SQL Server, or the physical performance of the server. As with any type of server, bottlenecks can occur because of pressure on one or more server resources such as memory, hard disk, input/output (I/O) devices, and central processing unit (CPU) usage. You need to be able to determine whether a system is improperly configured for the workload, or whether poor database design is the root cause of the problem. You need to proactively prevent or minimize problems and, when they occur, diagnose the cause and take corrective action.

Exam objectives in this chapter:

- Troubleshoot physical server performance
- Troubleshoot instance performance
- Troubleshoot database performance

Lessons in this chapter:

- Lesson 1: Troubleshooting Physical Server Performance 3
- Lesson 2: Troubleshooting Connectivity to a SQL Server Instance 28
- Lesson 3: Troubleshooting Database Performance 53

Before You Begin

To complete the lessons in this chapter, you must have completed the following tasks:

- Configured a Microsoft Windows Server 2003 R2 computer with SQL Server 2005 Enterprise Edition SP1 as detailed in the Appendix.
- Installed an updated copy of the AdventureWorks sample database as detailed in the Appendix.

No additional configuration is required for this chapter.

Real World*Ian Mclean*

When troubleshooting a SQL Server 2005 server problem, determining which resource is causing the bottleneck isn't sufficient; you also need to find out why the resource is under pressure. For example, when I was troubleshooting an underperforming server, I came across a CPU bottleneck. The results of previous monitoring showed that the CPU resource had not previously been under stress and the bottleneck had occurred suddenly and unexpectedly. Rather than rushing out to buy more CPUs, I investigated further and found that a nonoptimal query plan was causing a batch recompilation. I discuss query performance in Chapter 2, "Analyzing Queries."

Lesson 1: Troubleshooting Physical Server Performance

This lesson discusses the physical performance of the server on which SQL Server 2005 is installed, as well as the tools that you use to diagnose physical server problems. The principal tools you use for this purpose are the Windows Server 2003 Performance tools—Systems Monitor and Performance Logs and Alerts.

NOTE Service packs

The service pack level at the time of writing this book is Service Pack 1 (SP1). Unless otherwise indicated, all the information in the chapter applies to both SQL Server 2005 and SQL Server 2005 SP1.

You can also use SQL Server Profiler to troubleshoot physical server performance. When you have identified the physical resource under pressure, SQL Server Profiler can help you determine why that resource is under pressure. Lessons 2 and 3 of this chapter describe how you can use SQL Server Profiler in conjunction with the Database Engine Tuning Advisor (DTA), and dynamic management views (DMVs) to troubleshoot database-related problems.

Events related to SQL Server 2005 failures are written to the Windows Event log and the SQL Server log. Chapter 3, “Failure Diagnosis,” describes how you can access these logs to diagnose SQL Server 2005 failures.

After this lesson, you will be able to:

- Use System Monitor and Performance Logs and Alerts to identify hardware bottlenecks.
- Evaluate memory usage.
- Evaluate disk usage.
- Evaluate CPU usage.
- Evaluate network usage.
- Troubleshoot SQL Server 2005 connectivity issues.

Estimated lesson time: 60 minutes

Using System Monitor and Performance Logs and Alerts

In routine performance monitoring—described in Chapter 5, “Performance Monitoring”—you compare the results obtained with performance baselines to determine and track trends. In this chapter, however, we are concerned with discovering which resource (or resources) is under pressure and the counter values that indicate this pressure.

You can use System Monitor to get instant counter values and diagnose problems that result in unacceptable performance degradation, and you can set performance *alerts* to detect when a counter exceeds or falls below a predefined value. More typically, you diagnose reasons for performance degradation by creating a log and monitoring counters over a 24-hour period. This should be a normal 24-hour period—not a weekend or a holiday.

You can create *counter logs* that start immediately or at a specified time, and record the value of performance counters at predefined intervals. You can create alerts that send you a message, write events in the Application log, or start executable programs if a counter reading goes above or below a predefined value. You can create *trace logs* that record performance data whenever events related to their source provider occur. A *source provider* is an application or operating system service that has traceable events.

You can select the format in which log files are stored. Formats include comma-delimited text file, tab-delimited text file, binary file, binary circular file, and SQL database file. The facility to store logs as SQL database files is particularly useful when troubleshooting or monitoring a SQL Server 2005 server. You can display the log data as a graph, histogram, or report.

Initially, you would monitor the following counters:

- Memory: Pages/sec
- Memory: Available Bytes
- SQL Server: Buffer Manager: Buffer Cache Hit Ratio
- Physical Disk: Disk Reads/sec
- Physical Disk: Disk Writes/sec
- Physical Disk: % Disk time
- Physical Disk: Avg. Disk Queue Length
- Physical Disk: % Free Space
- Logical Disk: % Free Space
- Processor: % Processor Time
- System: Processor Queue Length
- Network Interface: Bytes Received/sec
- Network Interface: Bytes Sent/sec
- Network Interface: Output Queue Length
- SQL Server: General: User Connections

This list is a small subset of the counters that are available to you, but monitoring these counters will help you find many of the common and more obvious server-related performance problems.

BEST PRACTICES Run Performance Tools on the server you are troubleshooting.

Typically, you can run Windows Server 2003 Performance Tools on the SQL Server 2005 server you are troubleshooting. Other troubleshooting tools—for example, SQL Server Profiler—generate a larger performance hit and are best run from a monitoring server that has a fast, reliable connection to the server you are testing. If you want to determine the level of resources that the Performance Tools use, monitor the resource counters that let you specify the *smlogsvc* instance. *Smlogsvc* is the service that implements Performance Tools.

Evaluating Memory Usage

You identify memory bottlenecks through excessive *paging*, high memory consumption, a low buffer cache hit ratio, and a high volume of disk read and write I/O operations. The following list shows the counters you need to monitor:

- Memory: Pages/sec
- Memory: Available Bytes
- SQL Server: Buffer Manager: Buffer Cache Hit Ratio
- Physical Disk: Disk Reads/sec
- Physical Disk: Disk Writes/sec

Memory: Pages/sec

This counter measures the number of pages per second that are paged out from random access memory (RAM) to virtual memory on the hard disk. A high reading indicates that server memory is under stress. The more paging that occurs, the more I/O overhead the server experiences. Paging is a normal server operation that you need to keep to an acceptable level. You should not attempt to eliminate it.

Assuming that SQL Server is the only major application running on your server, the average value of this counter should be between zero and 20. Spikes greater than 20 are normal. If your server is averaging more than 20 pages per second, one of the more likely causes is a memory bottleneck. In general, the more RAM a server has, the less paging it has to perform.

NOTE Process: Page Faults/sec

To determine whether SQL Server or another process is the cause of excessive paging, monitor the Process: Page Faults/sec counter for the SQL Server process instance.

Memory: Available Bytes

SQL Server 2005 attempts to maintain from 4 to 10 MB of free physical memory. The Memory: Available Bytes counter measures the amount of free physical memory on a server. The average value of this counter should be greater than 5 MB; otherwise, your server could be experiencing a performance hit because of memory pressure.

BEST PRACTICES The /3GB switch

You can sometimes improve the performance of a SQL Server 2005 server by including the /3GB switch in the Boot.ini file. This enables SQL 2005 Server to use 3 GB of RAM. The effectiveness of this technique depends on how much RAM is available in your server and what other applications are running. For more details about memory management and the /3GB and /PAE switches in the Boot.ini file, access support.microsoft.com/kb/283037/en-us.

SQL Server: Buffer Manager: Buffer Cache Hit Ratio

This counter indicates how often SQL Server 2005 accesses the buffer rather than the hard disk to get data. If your server is running online transaction processing (OLTP) applications, this ratio should exceed 90 percent, and ideally it should be 99 percent (or more). A buffer cache hit ratio lower than 90 percent normally indicates that RAM pressure is seriously slowing SQL Server 2005 performance and you need to take appropriate action (typically, add more RAM). If the reading is between 90 percent and 99 percent, you might not have an actual problem, but additional RAM could improve performance. If your database is very large, however, you might not be able to get close to 99 percent, even if you put the maximum amount of RAM in your server.

Quick Check

- On a SQL Server member server, the average value of your Memory: Pages/sec counter is 30, the average value of your Memory: Available Bytes counter is 4 MB, and the average value of your SQL Server: Buffer Manager: Buffer Cache Hit Ratio counter is 89 percent. The server runs OLTP applications. However, it is also used as a file and print server. What counter

should you monitor to determine whether memory pressure is being caused by SQL Server operations?

Quick Check Answer

- Monitor the Process: Page Faults/sec counter for the SQL Server process instance. You should, however, also consider moving the file and print server function to another computer.

If your server is running online analytical processing (OLAP) applications, the buffer cache hit ratio can be significantly less than 90 percent because of how OLAP works. For this reason, you should treat the Buffer Cache Hit Ratio counter value with caution and use it for diagnostic purposes only if you already have evidence that SQL Server 2005 is accessing the hard disk more frequently than it should. On the other hand, adding extra RAM almost always improves the performance of a SQL Server 2005 server.

Real World

Ian McLean

Cynics on the Internet (and they abound) list procedures that artificially raise the value of the buffer cache hit ratio and use this information to discredit the use of the counter as a diagnostic tool. Here is one such procedure:

```
SQL>
SQL> -- Do NOT run this procedure in a production environment..
SQL> DECLARE
  2  v_dummy dual.dummy%TYPE;
  3  BEGIN
  4  FOR I IN 1..10000000 LOOP
  5  SELECT dummy INTO v_dummy FROM dual;
  6  END LOOP;
  7  END;
  8  /
```

I don't consider this to be a valid criticism. The procedure loops, retrieving the same data from RAM many times, and therefore increases the number of RAM accesses. It shouldn't stop you from using this valuable counter, even though the results are sometimes difficult to interpret.

Physical Disk: Disk Reads/sec and Physical Disk: Disk Writes/sec

The values of these counters can be difficult to interpret, because the acceptable transfer rate limits depend on the hardware installed in your server. For example, ultra-wide small computer system interface (SCSI) disks can handle from 50 through 70 I/O operations per second. Thus, the absolute values in the counters might not indicate that your disk resource is under threat, but a reading that grows over time can indicate increasing pressure on your memory resource. You should analyze the values of the Physical Disk: Disk Reads/sec and Physical Disk: Disk Writes/sec counters in conjunction with the Memory: Pages/sec, Memory: Available Bytes, and SQL Server: Buffer Manager: Buffer Cache Hit Ratio counters to determine whether your memory resource is under stress.

NOTE Sequential or random I/O

Whether the I/O is sequential or random can have a strong impact on values for disk reads per second and disk writes per second.

Evaluating Disk Usage

Disk counters are enabled by default on Windows Server 2003 servers. You can use both physical and logical disk counters to determine whether a disk bottleneck exists. You can partition a single physical disk into several *logical volumes*, although SQL Server 2005 servers more commonly use disk arrays. In general, the most useful counters for identifying pressure on the disk resource of a SQL Server 2005 server are those that monitor a physical array, rather than logical partitions or individual disks within the array.

In a production environment, SQL Server 2005 server operation can result in very large database and transaction log files, and disk capacity limits can be a potential problem, even with the very large disk arrays currently available. In this situation, it is good practice to configure an alert that warns you when disk usage exceeds a pre-defined level. The counters you need to monitor to identify disk array resource bottlenecks are as follows:

- Physical Disk: % Disk Time
- Physical Disk: Avg. Disk Queue Length

When you are configuring an alert, on the other hand, you will probably want to monitor individual physical disks or partitions. In this case, the counters you can use are the following ones:

- Physical Disk: % Free Space
- Logical Disk: % Free Space

Physical Disk: % Disk Time

This counter measures the pressure on a physical hard disk array (not a logical partition or individual disks in the array). If the value of this counter exceeds 55 percent for continuous periods of 10 minutes or more over during a 24-hour monitoring period, the SQL Server 2005 server might be experiencing a disk I/O bottleneck. If this happens only a few times over the 24-hour period, disk I/O pressure is not a major problem. If, on the other hand, it happens more than once per hour, a disk I/O problem exists and you need to address it.

Physical Disk: Avg. Disk Queue Length

This counter also measures the pressure on a physical hard disk array. If it exceeds 2 for each individual disk drive in an array (that is, a five-disk RAID array has a queue length of 10 or greater) for a period of 10 minutes or more, you might have a disk I/O bottleneck for that array. As with the Physical Disk: % Disk Time counter, if this happens occasionally in your 24-hour monitoring period, the pressure is not serious. If, however, it happens often, you might need to increase the I/O performance on the server as previously described.

You should use both the % Disk Time and the Avg. Disk Queue Length counters to help you decide whether your server is experiencing an I/O bottleneck. If you see many time periods where the % Disk Time is over 55 percent and when the Avg. Disk Queue Length counter is over 2 per physical disk, you can be confident the server is experiencing a disk I/O bottleneck.

Setting a Free Disk Space Alert

You can configure alerts on the Physical Disk: % Free Space and Logical Disk: % Free Space counters, and you can select the instances of these objects that correspond to the physical disks and logical partitions for which disk usage could be a problem. Typically, a value of 15 percent free disk space or less should trigger an alert.

IMPORTANT Performance log files can grow rapidly.

Counter and trace log files can grow rapidly and become very large, especially if you specify a short sample interval. Take care to place such files on a disk with sufficient free space; otherwise, the files could cause the very problem they are meant to detect.

Evaluating Processor Usage

You can evaluate the pressure on a single CPU or on the CPU resource on a symmetric microprocessor (SMP) server. You can also set alerts on CPU usage, but this is, arguably, of less value than setting alerts on disk usage. CPU usage is typically 100 percent when a service or an application starts, and a problem exists only if usage remains at this level for a significant amount time or if a high average usage is detected. You assess pressure on the processor resource by monitoring the Processor and System objects. The counters you typically use for this purpose are the following ones:

- Processor: % Processor Time
- System: Processor Queue Length

Processor: % Processor Time

Each central processing unit (CPU) in your SQL Server 2005 server is an instance of this counter, which measures the utilization of each individual CPU. The counter is also available for all the CPUs by selecting *_Total* as the instance. If the % Total Processor Time (*_Total*) counter exceeds 80 percent for continuous periods of 10 minutes or more during a 24-hour monitoring period, you might have a CPU bottleneck. If these busy periods occur only once or twice during the 24-hour period, the problem is probably not serious. If, on the other hand, they occur frequently (say, more than once per hour), you might need to reduce the load on the server or check query performance.

System: Processor Queue Length

You should monitor this counter in addition to the Processor: % Processor Time counter. If it exceeds 2 per CPU for continuous periods of 10 minutes or more during a 24-hour monitoring period, you might have a CPU bottleneck. If, for example, you have 3 CPUs in your server, the Processor Queue Length should not exceed 6 for the entire server.

Evaluating Network Usage

You can evaluate network usage by monitoring the Network Interface performance object counters. These counters measure the rates at which bytes and packets are sent and received over a TCP/IP connection. The first instance of the Network Interface object (Instance 1) represents the loopback, which is a local path through the protocol driver and the network adapter. All other instances represent installed network adapters.

Typically, no absolute maximum and minimum values exist for Network Interface counters (with the exception of the Output Queue Length counter, which should always be zero). Modern, high-speed network adapters and local area network connections can comfortably handle traffic levels that would overwhelm older hardware. You need to detect whether the traffic your network handles is increasing over time or has increased suddenly and unexpectedly. In the latter case, your network could be experiencing an external attack, and you might want to configure an alert to detect this. The level of traffic that triggers such an alert depends entirely on the typical level of traffic on your network. The counters you typically use to monitor network traffic are as follows:

- Network Interface: Bytes Received/sec
- Network Interface: Bytes Sent/sec
- Network Interface: Bytes/sec
- Network Interface: Output Queue Length

Network Interface: Bytes Received/sec

This counter shows the rate at which bytes (including framing characters) are received over each network adapter (or instance of the counter). A sudden, unexpected increase in this value could indicate that your network is experiencing an external attack.

Network Interface: Bytes Sent/sec

This counter shows the rate at which bytes (including framing characters) are sent over each network adapter (or instance of the counter). A sudden, unexpected increase in this value on a SQL Server 2005 server could indicate that a large volume of information is being accessed. If you cannot find an explanation for such a surge, further investigation is required.

Network Interface: Bytes/sec

This counter indicates the total level of network traffic both to and from a server. A sudden increase in the value in this counter could indicate an attack on your network—for example, a denial of service (DoS) attack. An increase in the average value in this counter over time could indicate that your network resource is coming under stress.

BEST PRACTICES Use Network Interface: Bytes/sec to trigger an alert.

If your company's Web site is linked to a SQL Server 2005 server, you should consider configuring an alert so that a sudden increase in the Network Interface: Bytes/sec counter on that server sends you a warning and triggers a Network Monitor capture.

Network Interface: Output Queue Length

This counter shows the length of the output packet queue in packets. If this is longer than two, a network bottleneck exists and should be found and eliminated. In modern network interface cards (NICs), the requests are queued by the Network Driver Interface Specification (NDIS) hardware, and the value of this counter should always be zero. In this case, a nonzero value could indicate faulty hardware.

Evaluating User Connections

The number of user connections affects SQL Server 2005 performance. If you are diagnosing performance problems, you should monitor the following counter:

- SQL Server: General Statistics: User Connections

SQL Server: General Statistics: User Connections

This counter shows the number of user connections, not the number of users currently connected to SQL Server 2005. If the value in this counter exceeds 255 for continuous periods of 10 minutes or more during a 24-hour monitoring period, you might have a bottleneck and should take corrective action.

Exam Tip Typically, an exam question presents you with a screen shot of the report view of a performance log, showing the average values for a number of counters. The question asks you to determine which resource is under pressure. You should, therefore, know which counters monitor which resources, and what the acceptable average values are—for at least the commonly used counters listed in this lesson.

Solving Resource Problems

When the values returned by your Performance counters indicate pressure on a resource, you should take action to clear the bottleneck and to solve any problems that might be causing it. Often, the professional response is not to add more memory, more hard disk resource, more processor power, or faster network connections, but rather to identify and solve the underlying problems. You might need to tackle resource problems in the following areas:

- Memory
- Disk I/O
- Physical drives and disk controllers
- Disk space
- CPU utilization
- Processor queues
- Network structure and bandwidth
- User connections

Memory

Adding more RAM usually alleviates the problems caused by excessive paging. I confess to being a RAM junkie, always looking for excuses to add more memory to my servers. However, it is unprofessional to leap at the first (expensive) solution, and you should perform further investigation before adding RAM. If you encounter a high pages/sec reading, you should next look at the SQL Server Buffer Manager: Buffer Cache Hit Ratio counter.

A SQL Server 2005 server should have a Buffer Cache Hit Ratio of 99 percent or higher. If your server has an average Buffer Cache Hit Ratio above 99 percent over a period of 24 hours and your average Pages/sec reading over the same period is greater than 20, check to see what applications other than SQL Server 2005 are running on the server. If possible, move these applications to another server. In general, SQL Server 2005 should be the only major application on a physical server.

If your SQL Server is not running any other major applications, an average Pages/sec reading greater than 20 could indicate that someone has changed the SQL Server memory settings. You should configure SQL Server with the Dynamically Configure SQL Server Memory option, and set the Maximum Memory setting to the highest level.

If the average value of the Memory Object: Available Bytes counter is less than 5 MB, the solution could again be to move any other major applications to another server, to change the SQL Server 2005 server's memory configuration settings, or to add more RAM.

Disk I/O

The first thing to check if your counter values indicate an excessive number of disk I/O operations is whether this situation is caused by a memory bottleneck, leading to excessive paging and a large number of disk reads and writes. If, however, you have a genuine disk I/O bottleneck, you can often solve the problem by reducing the load on the server (for example, by moving major applications other than SQL Server 2005 to another server).

Physical Drives and Disk Controllers

When a disk bottleneck is not related to a memory bottleneck and you cannot solve it by moving applications to another server, the problem probably is related to the physical drive or disk array or to the disk controller. You can reduce disk pressure on these resources by adding drives to the array, installing faster drives or a faster controller, adding cache memory to the controller card, or using a different version of *redundant array of independent disks (RAID)*.

Real World

Ian McLean

If I identify a bottleneck in the physical disk resource, I always check whether any new database applications are running before I squander some of my precious budget on new hardware. A database application that's not adequately indexed can put pressure on this resource. It always pays to check that your indexes are optimized. I discuss index maintenance and optimization in Chapter 2.

Sometimes you can alleviate pressure on a physical disk resource by performing an offline defragmentation of the files on the disk, resulting in contiguous files and more efficient information transfer. Although this is a low-cost solution in terms of hardware, it might be unacceptable because of the time that the disk needs to be taken offline—which could be considerable for a large disk array.

Hardware implementations of RAID that incorporate a battery backed-up write-back cache can speed up OLTP database applications and are, consequently, a popular choice for SQL Server 2005 servers. This type of hardware RAID implementation uses a specialized disk controller and typically implements its own specific monitoring and troubleshooting solutions, which are beyond the scope of this book.

Disk Space

If a disk, disk array, or logical volume is running short of disk space, you should identify large files that you can move to storage areas that have spare capacity. However, you should also check whether files are increasing in size, number, or both. If your disk storage holds *transaction log* files and if you have set the option for the transaction log files to grow automatically, find out when the last transaction file backup occurred. A backup truncates the transaction log files. This process does not reduce the physical file size, but SQL Server reuses this truncated, inactive space in the transaction log instead of permitting the transaction log to continue to grow and to use more space.

NOTE An alert can initiate a backup.

If transaction log file storage presents a serious problem, you can specify an alert that starts a program or batch file, which in turn starts a transaction file backup. The alert should also warn you that this is happening. If you want to reduce the physical size of the transaction log, you can use the DBCC SHRINKFILE Transact-SQL statement after the backup is complete. You should also consider reviewing your backup strategy.

MORE INFO Performance condition alerts

You can also use SQL Server *performance condition alerts*—for example, on the SQLServer:Databases: PercentLogUsed counter to trigger a transaction log backup and send a message to you by using Database Mail. For more information, search Books Online for “Defining Alerts” or access [msdn2.microsoft.com/en-us/library/ms180982\(d=ide\).aspx](http://msdn2.microsoft.com/en-us/library/ms180982(d=ide).aspx).

Performance log files grow over time and might end up creating the very bottleneck they were designed to detect. The maximum sizes of the Windows Event log and SQL Error log files are predefined, but these files can grow in number. You should consider a policy of regularly archiving log files to offline storage. However, data storage requirements almost inevitably grow over time, and you need to ensure that a budget is available for periodic expansion and upgrading of your disk arrays.

CPU Utilization

If both the Processor Queue Length and the % Total Process Time counters are exceeding their recommended values during the same continuous time periods, the CPU resource is under stress and you need to take the appropriate action. You can add CPUs, install faster CPUs, or install CPUs that have a larger on-board *level 2 (L2) cache*.

Real World

Ian McLean

On a number of occasions, I have had to decide whether to purchase a faster CPU with a smaller L2 cache or a slower CPU with a larger L2 cache. Unfortunately, the faster CPUs currently on the market have smaller caches, and vice versa. In general, if the SQL Server 2005 server for which I'm specifying the resource uses only one or two CPUs, I go for speed, with L2 cache as a secondary consideration, although I always specify the largest L2 cache I can for a given clock rate.

If, however, the server has four or more CPUs, I go for the CPUs with the largest L2 cache, even though their speed might not be as high. For SQL Server 2005 to run optimally on servers with four or more CPUs, the L2 cache has to be as large as possible; otherwise, much of the power of the additional CPUs goes to waste.

Fortunately, I've never been asked to specify CPUs for a server that uses three of them.

Processor Queues

A possible reason for the Processor Queue Length counter regularly exceeding its recommended maximum value is that an excessive number of worker threads are waiting to take their turn. Thus, if the value of this counter is high but the Processor: % Processor Time counter readings are not exceeding their limit, you should consider reducing the SQL Server Maximum Worker Threads configuration setting. You can configure this setting by using the Processors page in SQL Server Management Studio (SSMS) to modify the properties of your server. By reducing the maximum permitted number of worker threads, you either force thread pooling to start or force the processor to take greater advantage of this feature if it has already started.

Network Structure and Bandwidth

Network structures in a production environment are often complex, involving the use of routers and switches to implement subnets and *virtual local area networks (VLANs)*. Typically, a SQL Server 2005 server will be located on an internal corporate network protected by both a corporate and an Internet firewall, and by a proxy solution such as Microsoft Internet Security and Acceleration (ISA) server. Network management and network analysis using Microsoft Network Monitor or third-party sniffer tools is beyond the scope of this lesson, but you need to remember that just because you detected a network bottleneck using Performance counters on a SQL Server 2005 server, the server hardware is not necessarily the cause of the problem. Sudden and unexpected increases in network traffic can also indicate that the network is experiencing an internal attack. Although something most certainly needs to be done about this, it is seldom the responsibility of the DBA.

NOTE A network goes at the speed of its slowest component.

More than once, I have come across a network connected by gigabit Ethernet with the most modern switches, routers, and NICs specified, and I have found that a critical part of it passes through an old 10-megabit hub that everybody had forgotten about.

If, however, your Performance log indicates network pressure that is specific to your SQL Server 2005 server, the solution is to install faster NICs or share the tasks carried out by that server over one or more additional computers.

User Connections

If the value of the SQL Server: General Statistics: User Connections counter exceeds 255 for continuous periods of 10 minutes or more during a 24-hour monitoring period, try increasing the SQL 2005 Server configuration setting Maximum Worker Threads to a value higher than 255. If the number of connections exceeds the number of available worker threads, SQL Server will begin to share worker threads, which can degrade performance. The setting for Maximum Worker Threads should be higher than the maximum number of user connections your server ever reaches.

NOTE The Maximum Worker Threads setting

When the Processor Queue Length counter exceeds its maximum accepted value, one solution could be to reduce the Maximum Worker Threads setting. When the User Connections counter value exceeds 255, you can often improve performance by increasing the value of this same setting. Troubleshooting server performance is not an easy task, and it often involves obtaining the best balance between contradictory requirements.

PRACTICE **Generating Performance Counter Logs and Alerts**

In the following practice, you use the Windows Server 2003 Performance Tools to generate a counter log to monitor memory performance counters on the member server on your test network.

► **Practice 1: Generating a Counter Log**

In this practice, you generate a counter log and add the relevant counters. In a production environment, you would configure this log to run over a 24-hour period and set the counter sampling intervals accordingly. For convenience, this practice configures the log to run for 5 minutes with a sampling period of 10 seconds.

MORE INFO Extending this practice exercise

Because of space considerations, Practice 1 is an introduction to the Performance Log tool. You can expand the listed procedure by adding more counters or by adding performance objects rather than specific counters. You can also capture the data in a SQL Server database. To accomplish the latter task, you need to use the Data Source Wizard to define a data source. For more information, search for "How to: Define a Data Source Using the Data Source Wizard" in SQL Server 2005 Books Online or access msdn2.microsoft.com/en-us/library/ms174543.aspx.

1. Log in to your domain at your member server by using either your domain administrator account or an account that has been added to the sysadmin server role. (If you are using virtual machine software, log on to your domain and connect to your member server.)
2. From the Programs (or All Programs) menu, choose Administrative Tools, choose Performance, and then choose System Monitor. Click Add Counters (the + button), and display the Performance Object drop-down menu. Figure 1-1 shows some of the performance objects added by SQL Server. The name of your member server might differ from that shown in the figure, depending on your test lab setup.
3. Close the Add Counters dialog box.
4. Expand Performance Logs and Alerts. Select Counter logs.
5. From the Action menu, choose New Log Settings. Name the new log **MyCounter-Log**, and click OK. The MyCounterLog dialog box is shown in Figure 1-2.

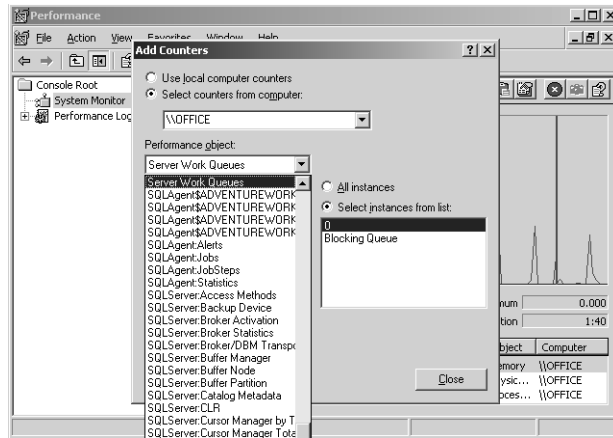


Figure 1-1 Performance objects added by SQL Server.

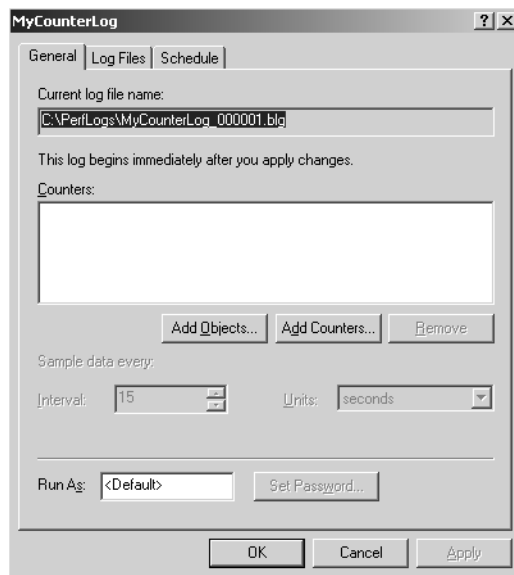


Figure 1-2 The MyCounterLog dialog box.

6. Click Add Counters. In the Add Counters dialog box, choose Use Local Computer Counters and then select the Memory performance object. Select Pages/sec (if not already selected), as shown in Figure 1-3. Click Add.
7. Select Available Bytes, and then click Add.
8. In the Performance Object drop-down list, select the SQL Server: Buffer Manager performance object, and add the Buffer Cache Hit Ratio counter. Click Close to close the Add Counters dialog box.

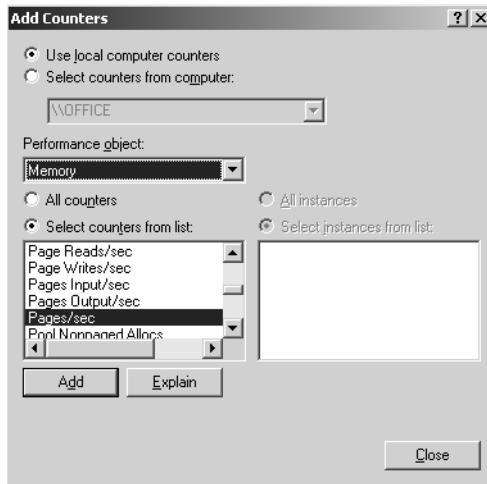


Figure 1-3 Selecting the Memory: Pages/sec performance counter.

9. On the General tab of the MyCounterLog dialog box, set the interval to 10 seconds. Type the name of the currently logged-in account in the Run As text box, as shown in Figure 1-4.

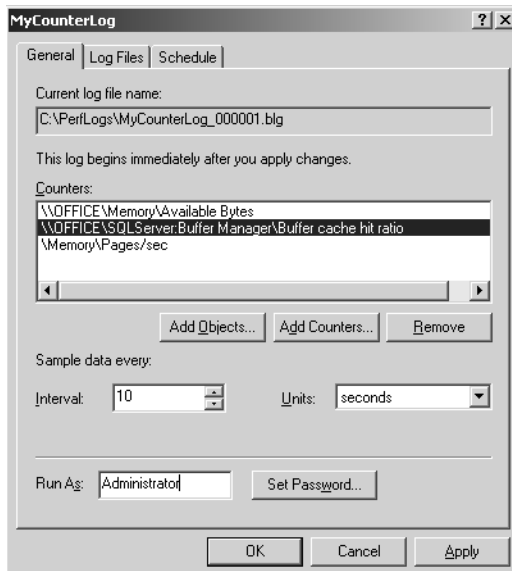


Figure 1-4 Setting the interval and run-as account.

10. On the Log Files tab, add a comment. Ensure that the log file type is set to binary (the default). If you want to expand this practice by specifying a SQL Server database and configuring a repository name, you can do so from this dialog box tab.

11. On the Schedule tab, configure the log to start manually and stop after five minutes, as shown in Figure 1-5.

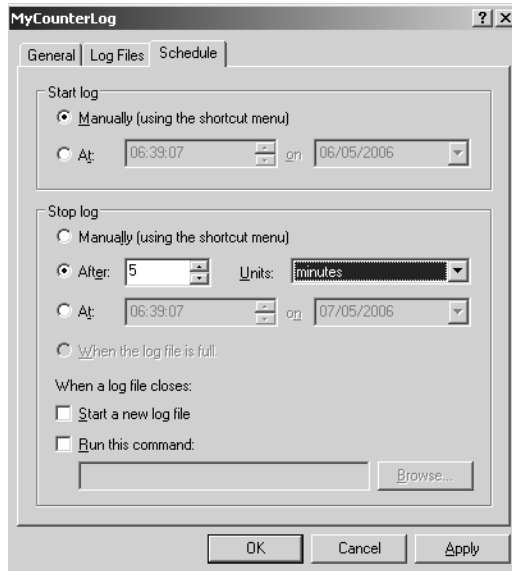


Figure 1-5 Scheduling the log.

12. Click OK. If a warning appears telling you that the folder you specified to hold the log files does not exist, click Yes to create this folder, and then click OK to create the log file.
13. MyCounterLog has been created but has not yet started. The log file symbol (shown in Figure 1-6) should be colored red to indicate this.

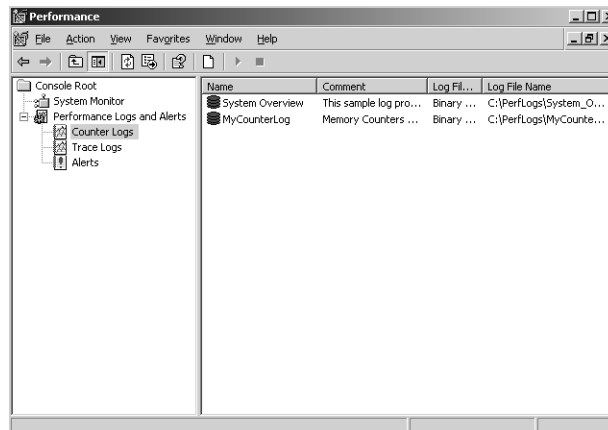


Figure 1-6 MyCounterLog created but not yet started.

14. Close the Performance console.

► Practice 2: Using a Counter Log

In this practice, you start the performance counter log you generated in Practice 1 and run a set of queries against the master and AdventureWorks databases on your member server. You can extend the procedure, if you want to do so, by running SQL Server applications that you suspect might be putting stress on your memory resource.

1. Log in to your domain at your member server, start the Performance console, and select Counter Logs, as you did in Practice 1.
2. In the console tree, expand Performance Logs And Alerts. Select the Counter Logs object. In the details pane, right-click MyCounterLog and choose Start.
3. From the Start menu, choose All Programs, Microsoft SQL Server 2005, SQL Server Management Studio. Connect to the database engine on your member server, specifying Windows Authentication (all defaults).
4. Click New Query as shown in Figure 1-7. This starts the Query Editor.

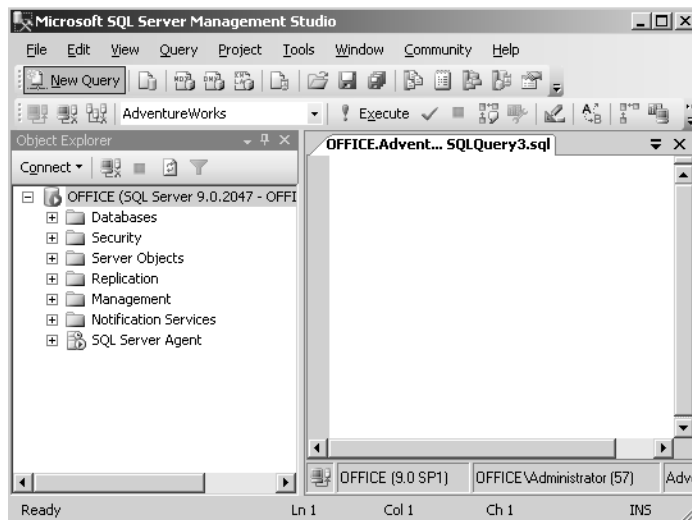


Figure 1-7 Starting the Query Editor.

5. In the Query pane, type the following text:

```
Use AdventureWorks  
EXEC sp_helpuser
```

Press F5 to run the query.

6. Run each of the following queries:

```
Use AdventureWorks  
EXEC sp_helprolemember 'db_owner'
```

Use AdventureWorks

```
SELECT * FROM sys.sql_logins
```

Use master

```
EXEC sp_helpuser
```

Use master

```
EXEC sp_helprolemember 'db_owner'
```

Use master

```
SELECT * FROM sys.sql_logins
```

7. In the Performance console, stop MyCounterLog if it has not already timed out.
8. Select System Monitor, and click View Log Data as shown in Figure 1-8.

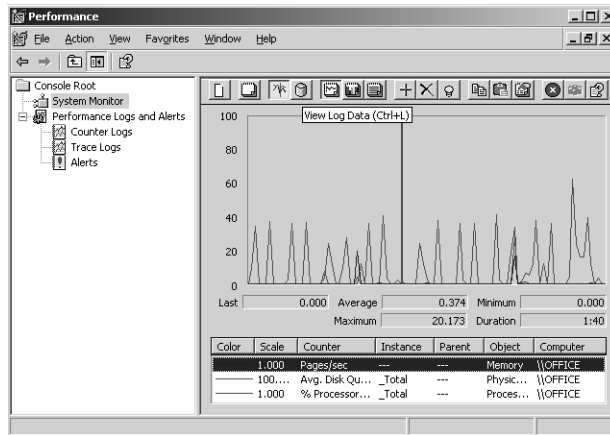


Figure 1-8 Selecting View Log Data.

9. On the Source tab on the System Monitor Properties dialog box, select Log Files and then click Add.
10. In the Look In drop-down list, browse to select the C:\perflogs folder. Select MyCounterLog_000001.blg and click Open.
11. On the Data tab, click Add. In the Add Counters dialog box, add the Memory: Available Bytes and SQL Server Buffer Management: Buffer Cache Hit Ratio counters. Click Close.
12. On the Data tab, remove the Processor and Physical Disk counters that are there by default, as shown in Figure 1-9. Select the three remaining counters in turn, and choose a different color for each if necessary. Click OK to close the System Monitor Properties dialog box.

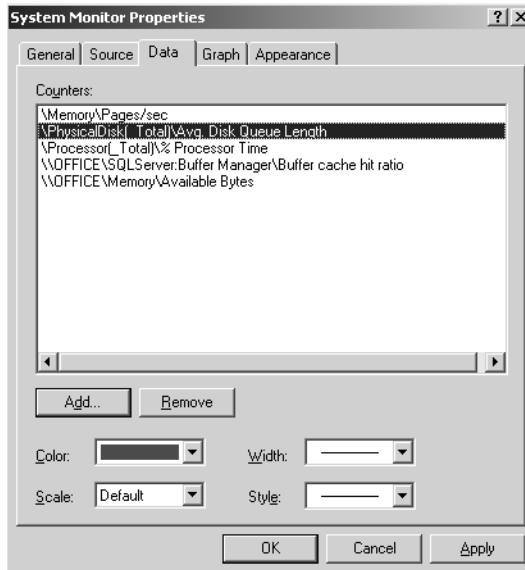


Figure 1-9 Removing unwanted counters.

- View the graph, histogram, and report pages. The report, shown in Figure 1-10, is typically the most useful view. A more comprehensive log file containing more counters and saved in SQL Server database format gives more information. In Figure 1-10, the value for Pages/sec is high, but the values for Available Bytes and Buffer Cache Hit Ratio are at acceptable levels. Memory is under stress, but this is not affecting SQL Server operations. Other applications are running on the member server.

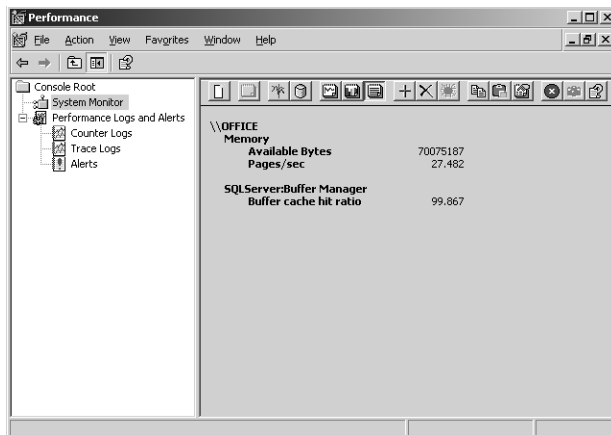


Figure 1-10 Report view.

Lesson Summary

- Windows Server 2003 Performance Tools are used to determine whether memory, disk space, disk I/O, processor, or network resources are under stress on a SQL Server 2005 server.
- You can generate counter and trace logs, set alerts, and use System Monitor to view counter values in real time or captured in a log file.
- Pressure on the memory resource can result from misconfigured server settings and badly written or incorrectly indexed queries. Memory stress can also result from other major applications running on a SQL Server 2005 server.
- Stress on RAM memory can result in stress on disk I/O. Free disk space should not be less than 15 percent of total disk space.
- Pressure on the processor resource can sometimes be relieved by moving non-SQL Server applications to other servers. Otherwise, you need to add more CPUs, faster CPUs, or CPUs with a larger L2 cache.
- The setting for Maximum Worker Threads should be higher than the maximum number of user connections your server ever reaches.

Lesson Review

You can use the following questions to test your knowledge of the information in Lesson 1, “Troubleshooting Physical Server Performance.” The questions are also available on the companion CD if you prefer to review them in electronic form.

NOTE Answers

Answers to these questions and explanations of why each answer choice is right or wrong are located in the “Answers” section at the end of the book.

1. Users in your company are reporting that queries and applications run against company databases on a SQL Server 2005 server are performing slowly. You want to make the best use of a limited hardware budget. You need to make decisions about which resources are most under stress during a typical 24-hour period. What should you do?
 - A. Create a series of performance alerts to notify you when the Memory: Bytes/sec, SQL Server: Buffer Manager: Buffer Cache Hit Ratio, Network Interface: Bytes Sent/sec, Network Interface: Bytes Received/sec, Processor: % Processor Time, and Physical Disk: % Disk Time counters exceed or fall below acceptable thresholds.

- B. Use System Monitor in graph mode to display information about the Memory: Bytes/sec, SQL Server: Buffer Manager: Buffer Cache Hit Ratio, Network Interface: Bytes Sent/sec, Network Interface: Bytes Received/sec, Processor: % Processor Time, and Physical Disk: % Disk Time counters. Perform this task at regular intervals during periods of peak server activity.
 - C. Use the Performance tab on Task Manager to monitor resources during periods of peak server activity.
 - D. Configure a counter log to capture information about the Memory: Bytes/sec, SQL Server: Buffer Manager: Buffer Cache Hit Ratio, Network Interface: Bytes Sent/sec, Network Interface: Bytes Received/sec, Processor: % Processor Time, and Physical Disk: % Disk Time counters. Schedule the log to sample the counter values at regular intervals, and store the information in either a comma-delimited text file or a SQL Server database file.
2. You administer a SQL Server 2005 server named SQLSRVA. Client applications that connect to SQLSRVA are responding slowly throughout the working day. You use System Monitor to check that the Pages/sec, %Processor Time, Disk Writes/sec, % Disk Time, and Avg. Disk Queue Length are at acceptable levels. What should you do next?
 - A. Create an alert that will warn you when available disk space falls below 15 percent.
 - B. Use System Monitor to determine the cache hit ratio.
 - C. Use System Monitor to examine the network usage counters.
 - D. Create a counter log to record the Available Bytes and Processor Queue Length counters over a 24-hour period.
3. You administer a SQL Server 2005 server named ServerA. Applications are running slowly on this server, and you create a counter log to monitor server resources. You find that ServerA has an average Buffer Cache Hit Ratio of 99.2 percent over a period of 24 hours and the average Pages/sec reading over the same period is 22.2. The server has a single processor. What should you do next?
 - A. Add more RAM.
 - B. Install a CPU with a larger L2 cache.
 - C. Check your disk counters.
 - D. Check what applications other than SQL Server 2005 are running on the server.

4. Your company Web site accesses a SQL Server 2005 server named SQL1. You suspect that SQL1 is experiencing periodic excessive network traffic caused by DoS attacks. You need to ensure that you receive a warning when such an attack occurs and that an appropriate audit trail is created. What should you do?
 - A. Create a trace log triggered by a high value in the Network Interface: Bytes/sec counter. Configure the log to record the Network Interface: Bytes Received and Network Interface: Bytes Sent counters.
 - B. Configure an alert triggered by a high value in the Network Interface: Bytes/sec counter that sends you a message and starts a Network Monitor capture.
 - C. Configure a counter log that monitors the Network Interface: Bytes Received and Network Interface: Bytes Sent counters over a 24-hour period.
 - D. Use Remote Desktop to monitor SQL1 from your workstation. Periodically check Task Manager to detect excessive network traffic.

Lesson 2: Troubleshooting Connectivity to a SQL Server Instance

Each *instance* (or installation) of SQL Server 2005 contains four system databases (master, model, msdb, and tempdb) and as many user databases as the DBA specifies. Each instance has its own ports, logins, and databases. In SQL Server 2005, you can have up to 50 instances running simultaneously on a server.

You can categorize instances as the *default (or primary) instance* and *named instances*. You can access the default instance using just the server name or its IP address—if you connect to a SQL Server 2005 server, you connect by default to its default instance. You access a named instance by appending a backslash and the instance name to the server name or its IP address.

If you have problems connecting to an instance or if queries against databases in this instance are running slowly, you need to troubleshoot resource bottlenecks and badly performing Transact-SQL statements within that instance.

After this lesson, you will be able to:

- Use System Monitor and Performance Logs and Alerts to identify resource bottlenecks related to instance performance.
- Use Profiler, the DTA, and DMVs to troubleshoot instance performance.
- Analyze tempdb database issues and tempdb contention.
- Evaluate instance memory usage.
- Analyze data and query cache issues.
- Analyze statement recompiles.
- Troubleshoot instance connectivity.
- Verify that the appropriate network libraries are specified.
- Analyze SQL CLR memory usage.
- Use MAXDOP statements to configure CPU parallelism.
- Troubleshoot waits and queues.

Estimated lesson time: 60 minutes

Troubleshooting Tools

In Lesson 1 of this chapter, “Troubleshooting Physical Server Performance,” we discussed the Windows Server 2003 Performance Tools: Systems Monitor and Performance Logs and Alerts. These tools are also used to monitor and troubleshoot

instance performance. In addition, you can use tools specific to SQL Server, including the following tools:

- SQL Server Profiler
- DTA
- DMVs

SQL Server Profiler

Microsoft SQL Server Profiler is a graphical user interface that enables you to trace events that occur in SQL Server 2005 and to capture and save data about each event to a file or table for analysis. An event is an action generated within an instance of the SQL Server database engine. The following events are examples of these:

- Login connections, failures, and disconnections
- Transact-SQL SELECT, INSERT, UPDATE, and DELETE statements
- Remote procedure call (RPC) batch status
- The start or end of a stored procedure
- The start or end of statements within stored procedures
- The start or end of a Transact-SQL batch
- An error written to the SQL Server error log
- A lock acquired or released on a database object
- An opened cursor
- Security permission checks

All the data generated by an event is displayed in the trace in a single row. This row is intersected by information columns that describe the event in detail. You can record and replay Profiler traces, and use the data to carry out the following tasks:

- Find slow-running queries and then determine what is making the queries run slowly.
- Step through statements to find the cause of a problem.
- Track a series of statements and then replay the trace on a monitoring server.
- Determine the cause of a deadlock.
- Determine what actions are using excessive CPU resource.
- Identify queries that are taking excessive time to process.

You can start Profiler from either Microsoft SQL Server 2005\Performance Tools on the All Programs menu or from within SSMS. You can select from the following information columns:

- **EventClass** An event class is a type of event that can be traced. The event class contains all the data that can be reported by an event. Examples of event classes are SQL:BatchCompleted, Audit Login, Audit Logout, Lock:Acquired, and Lock:Released.
- **EventCategory** An event category defines the way events are grouped within SQL Server Profiler. For example, all lock events classes are grouped within the Locks event category. However, event categories exist only within SQL Server Profiler. This term does not reflect the way database engine events are grouped.
- **DataColumn** A data column is an attribute of an event classes captured in the trace. Because the event class determines the type of data that can be collected, not all data columns are applicable to all event classes. For example, in a trace that captures the Lock:Acquired event class, the BinaryData column contains the value of the locked page ID or row, but the Integer Data column does not contain any value because it is not applicable to the event class being captured.
- **Template** A template defines the default configuration for a trace. Specifically, it includes the event classes you want to monitor with SQL Server Profiler. For example, you can create a template that specifies the events, data columns, and filters to use. A template is not executed, but rather is saved as a file with a .tdf extension. Once saved, the template controls the trace data that is captured when a trace based on the template is launched.
- **Trace** A trace captures data based on selected event classes, data columns, and filters.
- **Filter** When you create a trace or template, you can define criteria to filter the data collected by the event. To keep traces from becoming too large, you can filter them so that only a subset of the event data is collected.

Tracing a SQL Server Instance You can trace an instance of SQL Server by using either Profiler or system stored procedures. For example, you can create a Profiler trace to monitor exception errors. To do this, you select the Exception event class and the Error, State, and Severity data columns. You need to collect data for these three columns if the trace results are to provide meaningful data. You can then run a trace and collect data on any Exception events that occur. You can save trace data or use it immediately for analysis. You can replay traces at a later date, although you never

replay certain events, such as Exception events. You can also save the trace as a template to build similar traces in the future.

Running Profiler In a production environment, you typically run Profiler—and store its results—on a well-connected monitoring server. At a minimum, you should monitor the following:

- Events
 - Stored procedures: RPC Completed
 - TSQL: BatchCompleted
- Data Columns
 - TextData
 - Duration
 - CPU
 - Reads
 - Writes
 - ServerName and ApplicationName (if multiple servers are monitored)
 - EndTime
- Filters
 - Duration
 - System ID

For example, if you have a number of database applications running on a SQL Server instance and you configure a counter log that indicates disk activity on the volumes that contain the database data files is very high, you can find out what is causing high disk activity by capturing a Profiler trace and searching for events that have high read values or write values. The list just shown gives basic data about the instance you are tracing. If you are looking for a specific problem, you could add other events, and possibly other columns, to the list.

BEST PRACTICES Profiler templates

You can specify Profiler settings and save them as a template. If you require other DBAs to collect information regularly—for example, to satisfy the requirements of a service level agreement (SLA)—you can distribute a Profiler template for this purpose.

Database Engine Tuning Advisor (DTA)

DTA is an analysis tool that enables you to analyze the workload in single or multiple databases. DTA makes recommendations to alter the physical design structure of your SQL Server databases. These physical structures include clustered indexes, nonclustered indexes, indexed views, and partitioning. The purpose of DTA recommendations is to enable the query processor to execute the query in the least amount of time. DTA provides either a graphical user interface (GUI) or a command-line tool (dta).

After DTA has finished analyzing the workload, it generates a set of reports. You can use the DTA GUI to view these reports, or you can view the *Extensible Markup Language (XML)* output with any XML tool. DTA also provides a set of DMVs that return server state information. You can use this set to monitor the health of a server instance, diagnose problems, and tune performance.

BEST PRACTICES Run DTA on a test server

Both Profiler and DTA impose a heavy workload on a server. Profiler can test your production server from a well-connected monitoring server, but DTA requires a test server that duplicates some of the information on the production server. When using DTA, you do not need to import your entire database structure to your test server. DTA imports the metadata, statistics, and hardware parameters from the production server to the test server, and then performs tuning operations on the test server. You can implement the results on your production server during normal maintenance.

Dynamic Management Views (DMVs)

DMVs return server state information that you can use to monitor the health of a server instance, diagnose problems, and tune performance. There are two types of DMVs:

- **Server-scoped DMVs** These require the VIEW SERVER STATE permission on the server.
- **Database-scoped DMVs** These require the VIEW DATABASE STATE permission on the database.

To query a DMV, a user needs the SELECT permission on the object and the VIEW SERVER STATE or VIEW DATABASE STATE permission. This requirement enables the DBA to selectively restrict access of a user or login to DMVs. To do this, you first create the user in the master database and then deny that user SELECT permission on the DMVs that you do not want him or her to access. After this, the user cannot select

from these DMVs, regardless of the database context of that user. DMVs are organized into the following categories:

- Common Language Runtime Related
- I/O Related
- Database Mirroring Related
- Query Notifications Related
- Database Related
- Replication Related
- Execution Related
- Service Broker Related
- Full-Text Search Related
- SQL Operating System Related
- Index Related
- Transaction Related

Dynamic Management Functions

Dynamic management functions (DMFs) are related to DMVs and also return server state information. Like DMVs, they can be server scoped or database scoped. You define user permissions to access DMFs in the same way as for DMVs. DMFs are organized into the following categories:

- I/O Related
- Execution Related
- Index Related
- Transaction Related

Analyzing Tempdb Database Issues

The tempdb database stores internal and user objects, and it also stores the temporary tables, objects, and stored procedures that are created during SQL Server operation. A single tempdb database exists for each SQL Server instance, and tempdb issues can cause bottlenecks. Tempdb problems include lack of storage space and contention caused by excessive data definition language (DDL) operations and

excessive data manipulation language (DML) operations. Tempdb bottlenecks can cause unrelated applications running on the server to slow down or fail.

Tempdb Storage

The tempdb database contains user objects, internal objects, the version store, and free space. If you suspect that the database is running low on free space, you can monitor the SQL Server:Transactions: Free Space In Tempdb (kilobytes) performance counter and set an alert on this counter if free space drops below a critical value. You can also use the *sys.dm_db_file_space_usage* DMV to monitor disk space used by the user objects, internal objects, and version stores in the tempdb files. Additionally, you can use the *sys.dm_db_session_space_usage* and *sys.dm_db_task_space_usage* DMVs to identify large queries, temporary tables, or table variables that are using a large amount of tempdb disk space. For example, the following query returns the total number of free pages and total free space in megabytes available in all files in tempdb:

```
SELECT SUM(unallocated_extent_page_count) AS [free pages],  
(SUM(unallocated_extent_page_count)*1.0/128) AS [free space in MB]  
FROM sys.dm_db_file_space_usage;
```

Tempdb Contention

Contention occurs when more than one operation, process, or worker thread attempts to access the same resource at the same time. Sometimes the resource is locked. Sometimes a resource such as disk I/O or memory comes under stress and a bottleneck occurs.

Creating and dropping a large number of temporary tables and table variables can cause contention on metadata in tempdb. In SQL Server 2005, local temporary tables and table variables are cached to minimize metadata contention. However, for SQL Server to cache a table, no named constraints and no DDL operations (for example, CREATE INDEX and CREATE STATISTICS) can exist on the table after it is created. You can monitor the following performance counters for any unusual increase in temporary objects allocation:

- SQL Server: Access Methods: Workfiles Created /Sec
- SQL Server: Access Methods: Worktables Created /Sec
- SQL Server: Access Methods: Mixed Page Allocations /Sec
- SQL Server: General Statistics: Temp Tables Created /Sec
- SQL Server: General Statistics: Temp Tables For Destruction

If the contention in tempdb is caused by excessive DDL operation, you need to look at your application and see whether you can minimize this. If you use stored procedure scoped temporary tables, for example, can you move such tables outside of the stored procedure? If you can't, each execution of the stored procedure will cause a create/drop of the temporary table.

Monitoring Instance Memory Usage

The memory usage of a SQL Server instance typically grows to meet demand and optimization needs, and it can often rise quickly to several hundred megabytes. This type of increase affects the operation of the entire server. The quick fix for this situation is to open Windows Task Manager and determine which instance (or multiple instances) of SQL Server is using excessive RAM. You need to find out who is using the instance so that you do not end that user's session unannounced. As soon as it is safe to do so, locate the SQL Server instance in the Administrative Tools Services console on the SQL Server 2005 server, and then stop and restart it.

However, a quick fix is seldom a permanent solution. You need to find out why a particular instance is using excessive memory. If you know which SQL Server instance is using excessive memory and under what conditions, you can use Profiler to generate a trace, which could highlight problems with a specific database, a badly written query, or a rogue application.

Data Caching

SQL Server 2005 introduces a uniform caching framework that implements a clock algorithm. Currently, it uses two timers or *clock hands*—an internal clock hand and an external clock hand. The internal clock hand controls the size of a cache relative to other caches. It starts moving when the framework predicts that the cache is about to reach its upper limit.

The external clock hand starts to move when SQL Server as a whole is experiencing memory pressure. Movement of the external clock hand can be caused by external memory pressure, internal memory pressure, or both. Information about clock hand movements is exposed through the `sys.dm_os_memory_cache_clock_hands` DMV as shown in the following code:

```
SELECT *
FROM
    sys.dm_os_memory_cache_clock_hands
WHERE
    rounds_count > 0
    AND removed_all_rounds_count > 0
```

Each cache entry has a separate row for the internal and external clock hand. If you see increasing *rounds_count* and *removed_all_rounds_count* when an instance is running, data cache is under internal or external memory pressure.

Analyzing Statement Recompiles

Compilation is a significant part of a query's turnaround time, and SQL Server must compile each unique query at least once. The SQL Server 2005 database engine saves compiled query plans in a *query cache* for later use, thus saving recompilation time. However, when a batch or *remote procedure call (RPC)* is submitted to SQL Server, before it begins executing, the server checks for the validity and correctness of the query plan. If one of these checks fails, SQL Server might have to recompile the batch to produce a different query plan. Recompilations can also occur when SQL Server determines that there could be a more optimal query plan because of changes in underlying data. Compilations are CPU intensive, and excessive recompilations can result in a performance problem.

SQL Server 2005 introduces statement-level recompilation of stored procedures. When SQL Server 2005 recompiles stored procedures, it recompiles only the statement that caused the recompilation—not the entire procedure.

The SQL Server: SQL Statistics object provides counters to monitor compilation. You need to monitor the number of query compilations and recompilations in conjunction with the number of batches received to find out whether compiles are contributing to high CPU use. You should monitor the following counters:

- SQL Server: SQL Statistics: Batch Requests/sec
- SQL Server: SQL Statistics: SQL Compilations/sec
- SQL Server: SQL Statistics: SQL Recompilations/sec

If these counters indicate a high number of recompiles, you then need to look at the SP:Recompile and SQL: StmtRecompile event classes in a Profiler trace to determine which stored procedures SQL Server recompiles. The following list shows the key data columns you should monitor:

- EventClass
- EventSubClass
- ObjectID
- SPID (server process identifier)

- StartTime
- SqlHandle
- TextData

Of these columns, arguably the most useful is EventSubClass, which indicates the cause of the recompile. The column contains an integer from 1 through 11. Table 1-1 lists the integers and the corresponding causes for the SQL:StmtRecompile event class.

Table 1-1 SQL:StmtRecompile EventSubClass Integers

Integer	Cause of Recompilation
1	Schema changed
2	Statistics changed
3	Deferred compile
4	Set option changed
5	Temp table changed
6	Remote rowset changed
7	For Browse permissions changed
8	Query notification environment changed
9	Partition view changed
10	Cursor options changed
11	Option (recompile) requested

MORE INFO SQL:StmtRecompile event class

In SQL Server 2005, the preferred way to trace statement-level recompilations is to use the SQL:StmtRecompile event class rather than the SP:Recompile event class. For more information, look for "SQL:StmtRecompile Event Class" in Books Online, or access [msdn2.microsoft.com/en-us/library/ms179294\(SQL.90\).aspx](http://msdn2.microsoft.com/en-us/library/ms179294(SQL.90).aspx).

When you save a trace file, you can run a query to view all the recompile events that were captured in the trace.

NOTE Look for human intervention.

If you run Profiler by using the SQL:StmtRecompile or SP:Recompile event class and you detect a large number of recompiles, find out what the other members of your DBA team have been doing recently. A common reason that a large number of recompiles occurs is that a DBA executes UPDATE STATISTICS statements on all tables referenced by the most common stored procedures.

Configuring Connections

You can configure the maximum number of user connections for a SQL Server instance. Allowing excessive connections can result in stress on the memory resource. Applications connect by using network libraries, which pass packets between SQL Server 2005 servers and clients. Problems related to user and application connections are generally caused by misconfiguration.

User Connections

Each connection to an instance of SQL Server requires approximately 28 KB of memory overhead, regardless of whether the user is actually using the connection. By default, SQL Server 2005 allows a maximum of 32,767 user connections. SQL Server adjusts the maximum number of user connections automatically as needed, up to the maximum value allowable. For example, if only five users are logged in, SQL Server allocates five user connection objects. Therefore, you typically do not need to change the default setting.

However, in a large organization you could encounter problems from an excessive number of connections. Users running OLE DB applications need a connection for each open connection object, users running Open Database Connectivity (ODBC) applications need a connection for each active connection handle in the application, and users running DB-Library applications need one connection for each process started that calls the DB-Library dbopen function. You should estimate the number of connections based on system and user requirements, bearing in mind that on a system with many users, each user typically does not require a unique connection.

You set the maximum number of user connections by opening SSMS, connecting to the relevant server, right-clicking the server in Object Explorer, selecting Properties, and then selecting the Connections page. In the Maximum Number Of Concurrent Connections text box, you can type or select a value. The default setting is 0, which indicates an unlimited number of connections. You can also use the Transact-SQL

stored procedure *sp_configure* to change this setting. You must restart SQL Server for the change to take effect.

The Transact-SQL function @@MAX_CONNECTIONS returns the number of permitted simultaneous connections (not the number currently in use)—for example:

```
SELECT @@MAX_CONNECTIONS AS 'Max Connections'
```

Network Libraries

Open database connectivity (ODBC) and application connections to a SQL Server instance are implemented using network libraries, which pass packets between clients and SQL Server 2005 servers. Network libraries are implemented as *dynamic-link libraries (DLLs)* and perform all the operations required to activate specific interprocess communication (IPC) mechanisms. A network library exists for each network protocol you use—for example, SQL Server includes a TCP/IP network library and a named pipes network library.

At any given time, a server monitors multiple network libraries. By default, SQL Server 2005 Setup installs network libraries that you can configure to enable the server to listen on that library. You can change these configurations using the Server Network utility.

Microsoft recommends that you configure a SQL Server instance to use a dynamic port address by specifying a port address of 0 in SQL Server Configuration Manager. If this is not possible, use a port address of less than 1024.

An application (for example, Microsoft Visual Basic, C#, C++, or JScript) can specify the network library it uses by using the `SqlConnection.ConnectionString` property. An application that requires a connection verified by the Kerberos protocol requires you to configure SQL Server to use the TCP/IP network library, whereas one that uses NetBEUI requires the Named Pipes network library. If you do not specify a network and you use a local server, SQL Server uses the Shared Memory network library.

If a protocol is not used, you should disable its equivalent network library by using the Surface Area Configuration For Services And Connections tool or SQL Server Configuration Manager. This frees resources and reduces the possibility of an attack on the server. Errors associated with network libraries typically arise when a library is not enabled when it should be (or vice versa), when the port is misconfigured, or when a network library has stopped listening. For example, unauthorized disabling of a network library or a change of port can occur because of an attack on your network.

Dedicated Administrator Connections

In some error states, an instance of the database engine might not be able to accept new connections. This situation can prevent a DBA from connecting to the instance and diagnosing the problem. SQL Server 2005 introduces a *dedicated administrator connection (DAC)*. A member of the sysadmin fixed server role can use the new *sqlcmd* utility and the DAC to access and diagnose an instance of the database engine. The *sqlcmd* utility allows you to enter Transact-SQL statements, system stored procedures, and script files at the command prompt.

NOTE Log in interactively to create a DAC.

If you cannot connect to a SQL Server 2005 server that you want to administer by using SSMS, you can usually log in to the server interactively and create a DAC.

A problem can sometimes occur when a DBA can connect to the DAC on the default instance of a SQL Server 2005 server but cannot connect to the DAC endpoint on any of the named instances. In this case, you should verify that the instances are running, that client applications are able to access them, that the DAC is enabled for all instances, and that no other administrators are attempting to connect to the same DAC endpoints. In this situation, the solution is typically to start the *SQL Server Browser service* and configure this service to start automatically. The SQL Server Browser service listens for incoming requests for Microsoft SQL Server resources and provides information about SQL Server instances installed on the computer. SQL Server Browser performs the following tasks:

- Browses the list of available servers
- Connects to the correct server instance
- Connects to DAC endpoints

MORE INFO Using a DAC

For more information, search for "Using a Dedicated Administrator Connection" in Books Online or access [msdn2.microsoft.com/en-us/library/ms189595\(d=ide\).aspx](http://msdn2.microsoft.com/en-us/library/ms189595(d=ide).aspx).

SQL CLR Memory Usage

Programmers use the SQL common language runtime (CLR) integration feature to generate .NET assemblies. The feature is disabled by default, and you must enable it through the *sp_configure* stored procedure to use objects that are implemented using CLR integration.

Most memory used by the CLR comes from outside the SQL Server buffer pool. If the system runs low on memory, SQL Server aborts the CLR routine and memory is freed. Although this protects the server, it reduces whatever functionality the CLR routines were designed to implement, and you need to troubleshoot the situation to discover which routines are using excessive memory. To do this, you can monitor the garbage collection (GC) to identify the routines that the SQL Server service aborted. You should monitor the .NET CLR Memory object. Following is a list of the counters you need to check:

- **.NET CLR Memory: % Time In GC** This counter displays the percentage of elapsed time that the garbage collector has spent performing a garbage collection since the last garbage collection cycle. This counter usually indicates the work done by the garbage collector to collect and compact memory on behalf of the application. It is updated only at the end of every garbage collection. This counter is not an average; its value reflects the last observed value.
- **.NET CLR Memory: # GC Handles** This counter displays the current number of garbage collection handles in use. Garbage collection handles are handles to resources external to the common language runtime and the managed environment.
- **.NET CLR Memory: Allocated Bytes/Sec** This counter displays the number of bytes per second allocated on the garbage collection heap. It is updated at the end of every garbage collection, not at each allocation. This counter is not an average over time; it displays the difference between the values observed in the last two samples divided by the duration of the sample interval.

Quick Check

- What service connects to DAC endpoints on named instances?

Quick Check Answer

- The SQL Server Browser service.

Configuring CPU Parallelism

When SQL Server 2005 is running on a computer that has more than one CPU, such as a symmetric multiprocessor (SMP) computer, it automatically detects the best degree of parallelism for each instance of a parallel query execution or index data DDL operation. It does this based on a number of criteria:

- **Threads Available** The number of threads required increases with the degree of parallelism. If the thread requirement cannot be satisfied, the database engine

decreases the degree of parallelism automatically, or it abandons the parallel plan and executes a serial plan (single thread).

- **Type of query or index operation** A parallel plan operates most efficiently when it includes index operations that create an index, rebuild an index, or drop a clustered index; and queries that use CPU cycles heavily. For example, joins of large tables, joins of large aggregations, and sorts of large result sets are good candidates. The database engine compares the estimated cost of executing the query or index operation with the cost threshold for parallelism.
- **Number of rows to process** If the query optimizer determines that the number of rows is too low, it does not introduce exchange operators to distribute the rows, and the operators are executed serially.
- **Current distribution statistics** In earlier versions of SQL Server, the database engine abandoned parallel plans if the available statistics prevented it from providing the highest degree of parallelism. In SQL Server 2005, if the highest degree of parallelism is not possible, lower degrees are considered before the parallel plan is abandoned.

IMPORTANT Parallel index operations

Parallel index operations are available only in SQL Server 2005 Enterprise Edition.

When a query or index operation starts executing on multiple threads for parallel execution, the same number of threads is used until the operation is completed. The database engine re-examines the optimal number of thread decisions every time it retrieves an execution plan from the procedure cache. For example, one execution of a query can result in the use of a serial plan, a subsequent execution of the same query can result in a parallel plan using three threads, and a third execution can result in a parallel plan using four threads.

NOTE Dynamic cursors

Static and keyset-driven cursors can be populated by parallel execution plans. However, the behavior of dynamic cursors can be provided only by serial execution. The query optimizer always generates a serial execution plan for a query that is part of a dynamic cursor.

MAXDOP Statements

You can use the Max Degree Of Parallelism server configuration option to limit the number of processors SQL Server uses in parallel plan execution. You can override

the Max Degree Of Parallelism option for an individual query and index operation statements by specifying the MAXDOP query hint or MAXDOP index option. MAXDOP provides more control over individual queries and index operations. For example, you can use the MAXDOP option to control the number of processors SQL Server dedicates to an online index operation. In this way, you can balance the resources the server uses by an index operation with those of the concurrent users.

You can troubleshoot stress on the CPU resource by selecting multiple CPU instances of the processor object in (for example) the performance counter Processor: %Processor Time. However, it is probably more useful to run a query that involves (say) a large table join and then capture a Profiler trace.

Monitoring Waits and Queues

Waits are delays caused when a process or worker thread cannot immediately access the resources it requires because of excessive activity or inadequate resources. Resource waits occur when a worker thread requests access to a resource that is not available. Examples of resource waits are locks, latches, network, and disk I/O waits. Lock and latch waits are waits on synchronization objects.

Queue waits occur when a worker thread is idle, waiting for work to be assigned. You most typically see queue waits with system background tasks such as the deadlock monitor. External waits occur when a worker thread is waiting for an external event, such as an extended stored procedure call or a linked server query, to finish. When you diagnose blocking issues, remember that external waits do not always imply that the worker is idle, because the worker might actively be running some external code. The SQLServer:Wait Statistics performance object contains performance counters that report information about wait status. This object provides the following counters:

- Lock Waits
- Log Buffer Waits
- Log Write Waits
- Memory Grant Queue Waits
- Network IO Waits
- Non-Page Latch Waits
- Page IO Latch Waits
- Page Latch Waits

- Thread-Safe Memory Objects Waits
- Transaction Ownership Waits
- Wait For The Worker
- Workspace Synchronization Waits

Each counter contains the following instances:

- **Average Wait Time (ms)** Average time for the selected type of wait
- **Cumulative Wait Time (ms) Per Second** Aggregate wait time per second for the selected type of wait
- **Waits In Progress** Number of processes currently waiting on the selected type of wait
- **Waits Started Per Second** Number of waits started per second of the selected type of wait

Using the *sys.dm_tran_locks* DMV

Locks are managed internally by a part of the database engine called the lock manager. The *sys.dm_tran_locks* DMV returns information about currently active lock manager resources. Each row represents a currently active request to the lock manager for a lock that has been granted or is waiting to be granted.

The columns in the result set are divided into two main groups—resource and request. The resource group describes the resource on which the lock request is being made, and the request group describes the lock request.

The *sys.dm_tran_locks* DMV is useful in situations where, for example, an application running on a particular instance stops responding at random intervals; your processor, network, and disk utilization are within acceptable limits; and you suspect that locking could be causing a problem and need to examine additional information.

SQLServer:Locks Counters

The SQLServer:Locks object provides information about SQL Server locks. You can monitor multiple instances of the Locks object at the same time, with each instance representing a lock on a resource type. The following list shows the SQL Server:Locks counters:

- **Average Wait Time (ms)** Average amount of wait time (in milliseconds) for each lock request that resulted in a wait

- **Lock Requests/sec** Number of new locks and lock conversions per second requested from the lock manager
 - **Lock Timeouts (Timeout > 0)/sec** Number of lock requests per second that timed out, but excluding requests for NOWAIT locks
 - **Lock Timeouts/sec** Number of lock requests per second that timed out, including requests for NOWAIT locks
 - **Lock Wait Time (ms)** Total wait time (in milliseconds) for locks in the last second
 - **Lock Waits/sec** Number of lock requests per second that required the caller to wait
 - **Number of Deadlocks/sec** Number of lock requests per second that resulted in a deadlock
-

MORE INFO Activity Monitor

You can also use the Activity Monitor, which you access in the Object Explorer in SSMS, to diagnose problems with locks and user connections and to terminate a deadlocked or otherwise unresponsive process. However, you typically use this tool to troubleshoot database problems rather than instance problems. Lesson 3 of this chapter, “Troubleshooting Database Performance,” provides more information about Activity Monitor.

Monitoring Latch Wait Times

The SQLServer:Latches object provides counters to monitor internal SQL Server resource locks called *latches*. Monitoring the latches to determine user activity and resource usage can help you to identify performance bottlenecks. SQL Server 2005 provides three counters that you can use to measure latch activity:

- **Average Latch Wait Time (ms)** The time (in milliseconds) that latch requests have to wait before being granted. This counter does not include time for latches that do not have to wait.
- **Latch Waits/sec** The number of latch requests that could not be granted immediately.
- **Total Latch Wait Time (ms)** The total wait time (in milliseconds) for latch requests in the last second.

As with most performance logging solutions, monitoring wait times and latch wait times depends on comparing the current information against a baseline. Increasing

wait times and latch wait times indicate potential problems, typically pressure on memory or disk I/O.

If you detect a rise in latch activity, you should review the buffer cache hit ratio. If this counter is below 90 percent, you might need to add more RAM. If the ratio is above 90 percent, you should check disk queue lengths to determine whether excessive disk queuing indicates inadequately sized or badly configured disk subsystems.

If you determine that latch wait problems are occurring, you can use a SQL Server 2005 DMV to help identify the activity that is causing the issue. The DMV should include the following features:

- ***sys.dm_os_latch_stats*** Returns information, organized by class, about all latch waits
- ***sys.dm_os_wait_stats*** Returns information about the waits encountered by threads that are in execution
- ***sys.dm_db_operational_stats*** Returns current low-level I/O, locking, latching, and access method activity for each partition of a table or index in the database

Often, a particular activity is the cause of the performance problems. Instead of simply adding more RAM or disk resource, you can examine the process causing the issue. To track down the offending process, you must examine the relative wait numbers and wait times for each of the different latch classes to understand the performance of its SQL Server instances. This information can then be used to resolve or reduce the latch contention.

MORE INFO Locking in the database engine

For more information about locks, look up "Locking in the database engine" in SQL Server 2005 Books Online, or access msdn2.microsoft.com/en-US/library/ms190615.aspx.

PRACTICE Using SQL Server Profiler

This practice session introduces the Profiler tool. In a production environment, you capture a Profiler trace while running a database application or a complex query that stresses a server or a SQL Server instance. In this practice session, you capture a trace while running simple queries and extracting information from a DMV.

► Practice 1: Generating a Profiler Trace

In this practice, you run Profiler on your member server. In a production environment, you should run the tool on another machine and connect to the SQL Server 2005 server that is running your application. However, running Profiler on a domain controller is not a good idea. Therefore, on your test network, you use the same computer to host the SQL Server instance and run Profiler. To generate a Profiler trace, follow these steps:

1. Log in to your domain at your member server by using either your domain administrator account or an account that has been added to the sysadmin server role. If you are using virtual machine software, log in to your domain and connect to your member server.
2. Create a folder on your server to store Profiler traces—for example, C:\ProfileTrace.
3. From the All Programs menu, choose Microsoft SQL Server 2005, choose Performance Tools, and then choose SQL Server Profiler. From the File menu, choose New Trace. In the Connect To Server dialog box, click Options. On the Connection Properties tab, specify AdventureWorks as the database to which you want to connect and TCP/IP as the network protocol, as shown in Figure 1-11. Click Connect.

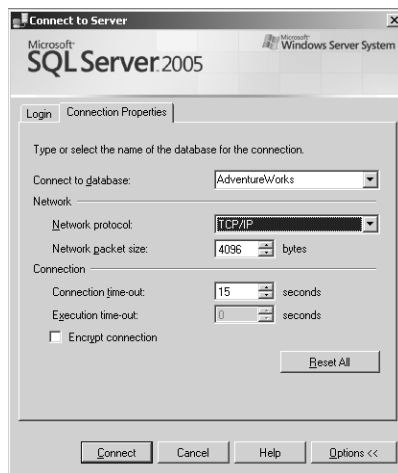


Figure 1-11 Making a Profiler connection.

4. Give the trace a name—for example, **MyTrace**.

5. On the Events Selection tab of the Trace Properties dialog box, you can select the columns and events you want to view. If you were tracing a large production operation, you would select specific columns and events. For the purpose of this practice, you will collect all available information. Select the Show All Events and Show All Columns check boxes. Click Run.
6. If necessary, open the Performance console. In the console tree, expand Performance Logs And Alerts and select the Counter Logs object. In the details pane, right-click MyCounterLog and choose Start.
7. If necessary, open SQL Server Management Studio. Connect to the database engine on your member server, specifying Windows Authentication. Connect to the default instance of SQL Server and specify the AdventureWorks database.
8. Click New Query to start the Query Editor.
9. In the Query pane, type the following text:

```
Use AdventureWorks
EXEC sp_helpuser
```

Press F5 to run the query.

10. Run each of the queries listed.

```
Use AdventureWorks
EXEC sp_helprolemember 'db_owner'
```

```
Use AdventureWorks
SELECT * FROM sys.sql_logins
```

```
Use AdventureWorks
SELECT *
FROM
sys.dm_os_memory_cache_clock_hands
WHERE
rounds_count > 0
AND removed_all_rounds_count > 0
```

11. In Profiler, from the File menu, choose Stop Trace, and then save it as a trace file in the folder you created.
12. You can use Profiler to view the trace and search for strings. Figures 1-12 through 1-15 show columns in the trace. Look for events that create a large number of disk reads and writes or have long durations. The LoginName column lets you know which users are running specific queries against a database.

EventClass	TextData
Trace Start	
ExistingConnection	-- network protocol: LPC set quoted...
ExistingConnection	-- network protocol: TCP/IP set quo...
ExistingConnection	-- network protocol: TCP/IP set quo...
ExistingConnection	-- network protocol: TCP/IP set quo...
ExistingConnection	-- network protocol: TCP/IP set quo...
ExistingConnection	-- network protocol: TCP/IP set quo...
SQL:BatchStarting	use adventureworks exec sp_helpuser
SQL:BatchCompleted	use adventureworks exec sp_helpuser
SQL:BatchStarting	use adventureworks exec sp_helprole...
SQL:BatchCompleted	use adventureworks exec sp_helprole...
SQL:BatchStarting	use adventureworks select * from sy...
SQL:BatchCompleted	use adventureworks select * from sy...
SQL:BatchStarting	use adventureworks select * from ...
SQL:BatchCompleted	use adventureworks select * from ...

Figure 1-12 EventClass and TextData columns.

ApplicationName	NTUserName	LoginName
SQLAgent - G...	Administrator	NWTRADERS\Administrator
Microsoft SQ...	Administrator	OFFICE\Administrator
Microsoft SQ...	Administrator	OFFICE\Administrator
Microsoft SQ...	Administrator	OFFICE\Administrator
Microsoft SQ...	Administrator	OFFICE\Administrator
	Administrator	OFFICE\Administrator
	Administrator	OFFICE\Administrator
	Administrator	OFFICE\Administrator
	Administrator	OFFICE\Administrator
	Administrator	OFFICE\Administrator
	Administrator	OFFICE\Administrator
	Administrator	OFFICE\Administrator
	Administrator	OFFICE\Administrator
	Administrator	OFFICE\Administrator
	Administrator	OFFICE\Administrator
	Administrator	OFFICE\Administrator

Figure 1-13 ApplicationName, NTUserName, and LoginName columns.

The screenshot shows a table with the following columns: CPU, Reads, Writes, Duration, ClientProcessID, and SPID. The data is as follows:

CPU	Reads	Writes	Duration	ClientProcessID	SPID
				1776	51
				3196	52
				3196	53
				2676	55
				2676	56
				2676	57
				2676	57
				2676	57
340	684	2	3707	2676	57
				2676	57
160	180	0	600	2676	57
				2676	57
70	105	0	616	2676	57
				2676	57
				2676	57
0	32	0	197	2676	57

Trace is stopped. Ln 16, Col 1 Rows: 16

Figure 1-14 CPU, Reads, Writes, Duration, ClientProcessID, and SPID columns.

The screenshot shows a table with the following columns: StartTime, EndTime, and BinaryData. The data is as follows:

StartTime	EndTime	BinaryData
2006-05-08 00:57:02...		
2006-05-05 03:13:45...		
2006-05-06 02:42:38...		
2006-05-06 02:43:39...		
2006-05-07 22:34:00...		
2006-05-07 22:35:24...		
2006-05-07 22:35:59...		
2006-05-08 01:05:00...		
2006-05-08 01:05:00...	2006-05-08 01:05:04...	
2006-05-08 01:05:32...		
2006-05-08 01:05:32...	2006-05-08 01:05:33...	
2006-05-08 01:06:06...		
2006-05-08 01:06:06...	2006-05-08 01:06:07...	
2006-05-08 01:06:21...		
2006-05-08 01:06:21...	2006-05-08 01:06:21...	

Trace is stopped. Ln 16, Col 1 Rows: 16

Figure 1-15 StartTime, EndTime, and BinaryData columns.

Lesson Summary

- You can use Windows Server 2003 Performance Tools, Profiler, DTA, and DMVs to analyze and troubleshoot instance performance.

- A single tempdb database exists for each SQL Server instance, and tempdb issues can generate bottlenecks.
- You can configure the maximum number of user connections to a SQL Server instance. Allowing excessive connections can result in stress on the memory resource.
- Microsoft recommends that you configure a SQL Server instance to use a dynamic port address by specifying a port address of 0 in the Server Network utility. If this is not possible, use a port address of less than 1024.
- You can control the maximum degree of CPU parallelism for individual query and index operation statements by specifying the MAXDOP query hint or MAXDOP index option.
- You can monitor waits (for example, lock waits, queue waits, latch waits, and I/O waits) by using the SQLServer:Wait Statistics and SQLServer:Latches performance objects.

Lesson Review

You can use the following questions to test your knowledge of the information in Lesson 2, “Troubleshooting Connectivity to a SQL Server Instance.” The questions are also available on the companion CD if you prefer to review them in electronic form.

NOTE Answers

Answers to these questions and explanations of why each answer choice is right or wrong are located in the “Answers” section at the end of the book.

1. You are a DBA employed by BlueSky Airlines. Users report that a booking application running on a SQL Server 2005 instance is performing poorly. After investigating the problem, you discover that disk I/O activity on the volume that contains the database data files is very high. You generate a Profiler trace while the application is running. How do you find out what SQL Server operations are causing the high disk activity?
 - A. Search for events that have high CPU activity.
 - B. Search for events that have high read values or high write values.
 - C. Search for events that have high duration values.
 - D. Search for events with an SPID value of 56.

2. You are a DBA employed by Coho Vineyards. Users report that a SQL Server 2005 stock control application stops responding for several seconds at random times throughout the day and affects many users at once. You cannot detect excessive pressure on the physical server resources and suspect that locks might be an issue. You need more information. What should you do?
 - A. Examine the *sys.dm_db_task_space_usage* DMV.
 - B. Examine the *sys.dm_tran_locks* DMV.
 - C. Examine the *sys.dm_os_memory_cache_clock_hands* DMV.
 - D. Run Profiler to examine the SQL: StmtRecompile event class.
3. You are the senior DBA at Northwind Traders. Your database administration team monitors a large number of SQL Server 2005 servers located at company headquarters and branch locations throughout the United States. Members of your team need to ensure that query response times comply with the requirements of Northwind Traders' service level agreements (SLAs). You need to provide your team with a consistent way of monitoring query response times on the servers. What should you do?
 - A. Instruct the administrators to use the Performance console to monitor the SQL Server Buffer Manager: Buffer Cache Hit Ratio counter.
 - B. Run Profiler by using the SQL: StmtRecompile event class. Distribute the Profiler trace to your team and instruct them to use DTA on their servers and follow the recommendations.
 - C. Create SQL Server Profiler templates that include query start times and end times. Distribute these templates to the members of your team.
 - D. Instruct your team to use the *sys.dm_db_session_space_usage* and *sys.dm_db_task_space_usage* DMVs.

Lesson 3: Troubleshooting Database Performance

SQL Server 2005 introduces SQL Server Management Studio, which provides a consistent and (arguably) user-friendly interface for managing and troubleshooting database issues. SSMS is an interactive, graphical tool that enables a DBA to write queries, execute multiple queries simultaneously, view results, analyze the query plan, and receive assistance to improve the query performance. The Execution Plan options graphically display the data retrieval methods chosen by the SQL Server query optimizer.

Specific troubleshooting tools such as System Monitor, Performance Logs and Alerts, SQL Server Profiler, and the DTA are also relevant to database troubleshooting, in addition to server and instance troubleshooting. SQL Server 2005 provides DMVs, stored procedures, Transact-SQL statements, and Database Console Commands (DBCCs), which you can use to isolate database problems.

After this lesson, you will be able to:

- Resolve space issues.
- Update statistics.
- Evaluate index usage.
- Defragment and rebuild indexes.
- Create missing indexes and drop inappropriate indexes.
- Audit and analyze poorly written queries.
- Monitor transaction log size and database growth.
- Investigate locking and resolve deadlocks.
- Determine optimum RAID configuration.

Estimated lesson time: 50 minutes

Resolving Space Issues

When a designer creates a database, he or she should define that database by using an adequate file size. Generous initial file sizes and adequate disk space reduce the chance that databases will run out of space and cause performance to degrade temporarily while they resize. The file size specified for a database depends on the estimated number of subscribers, subscriptions, events, and notifications that the instance and the application will support.

The DBA, on the other hand, is concerned with administering databases and resolving issues related to database size and out-of-space issues. Nevertheless, to understand why a database can experience space issues, you need to know about database structure.

An application database can store subscriptions, events, notifications, application status information, historical data in the form of chronicles, and metadata about the application. Event and notification data is removed according to a predefined schedule and data retention age. If you add data to a database, space issues do not involve only that data. Space issues also involve metadata indexes associated with the data, and event and notification information associated with applications and queries that use the data. For example, application databases typically contain from 5 to 10 MB of metadata.

The data in an instance database is more stable and typically grows more slowly than data in an application database. The size of the data in the database depends on the number of subscribers, the size of each subscriber row, and the number of subscriber devices. An instance database typically contains approximately 2 MB of metadata.

MORE INFO Instance and application databases

For more information about application and instance databases, their contents, and typical sizes, search for "Database Resource Planning" in Books Online or access [msdn2.microsoft.com/en-us/library/ms166482\(d=ide\).aspx](http://msdn2.microsoft.com/en-us/library/ms166482(d=ide).aspx).

You can configure a database to *auto-grow*, or you can grow it manually by using the ALTER DATABASE Transact-SQL statement. The advantages of using auto-grow are that you avoid the tedium of growing databases on a regular basis, and users are less likely to encounter out-of-space problems. The disadvantages are that you can run out of disk space and that auto-grow uses a considerable amount of system resource.

Real World*Ian Mclean*

The disk space issues of auto-grow are usually underestimated. Typically, a database auto-grows by a factor of 10 percent—a serious consideration when a 10-GB database grows by 1 GB. However, this also means you need an additional gigabyte of storage space to back up the database, additional space for transaction logs, and additional storage space to back up the transaction logs. My approach is to leave auto-grow enabled but monitor large and important production databases regularly. If they need to auto-grow, so be it—but not during peak business hours. A good tip is to have a few large but unimportant files in the database that you can delete to create space in an emergency (for example, a database that contains only a small amount of static information retrievable

from CD-ROM, but for which a large file size has been specified). Using DBCC SHRINKDATABASE with the NOTRUNCATE option frees space within the database without releasing it to the operating system. It also returns the current size of the files in the database (in 8-KB pages).

Monitoring Auto-Grow and Auto-Shrink

You can use Profiler to record events as they occur in an instance of the database engine. The recorded events are instances of the event classes in the trace definition. In SQL Server Profiler, event classes and their event categories are available on the Events Selection tab of the Trace File Properties dialog box. You use the Database event category to monitor auto-grow and auto-shrink events in a database. Following is a list of event classes of interest in this context:

- **Data File Auto Grow Event Class** This event indicates that the data file grew automatically. It is not triggered if you grow the data file explicitly through ALTER DATABASE.
- **Data File Auto Shrink Event Class** This event indicates that the data file has been shrunk.
- **Log File Auto Grow Event Class** This event indicates that the log file grew automatically. It is not triggered if the log file is grown explicitly through ALTER DATABASE.
- **Log File Auto Shrink Event Class** This event indicates that the log file shrunk automatically. It is not triggered if the log file shrinks explicitly through ALTER DATABASE.

Updating Statistics

After a series of INSERT, DELETE, or UPDATE Transact-SQL statements are performed on a table, the statistics might not reflect the true data distribution in a given column or index. If a column in a table has undergone substantial update activity since the last time the statistics were created or updated, SQL Server automatically updates the statistics by sampling the column values (using auto-update statistics).

The statistics auto-update is triggered by query optimization or by execution of a compiled plan, and it involves only a subset of the columns referred to in the query. If the query optimizer needs a particular statistics object when a query is first compiled and that statistics object exists, the statistics object is updated (if it is out of date). When

you execute a query and its plan is in the cache, the query optimizer checks the statistics on which the plan depends to see whether they are out of date. If so, SQL Server removes the plan from the cache and updates the statistics during recompilation of the query. SQL Server also removes the plan from the cache if any of the statistics that it depends on have changed.

The auto-update statistics feature can be turned off at different levels. At the database level, you can disable auto-update statistics using `ALTER DATABASE dbname SET AUTO_UPDATE_STATISTICS OFF`. At the table level, you can disable auto-update statistics using the `NORECOMPUTE` option of the `UPDATE STATISTICS` or `CREATE STATISTICS` command.

NOTE Sampling rate

Auto-statistics update is always performed by sampling the index or table using the default sampling rate. To set the sampling rate explicitly, you need to run `CREATE STATISTICS` or `UPDATE STATISTICS`.

The *sp_autostats* Stored Procedure

You can use the stored procedure *sp_autostats* to display and change the auto-update statistics setting for a table, an index, or a statistics object. You can re-enable the automatic updating of statistics by using `ALTER DATABASE`, `UPDATE STATISTICS`, or the *sp_autostats* stored procedure. You cannot override the database setting of `OFF` for auto-update statistics by setting it `ON` at the statistics object level.

NOTE Disabled nonclustered indexes

Statistics on disabled nonclustered indexes are also updated by *sp_updatestats*. The *sp_updatestats* stored procedure ignores tables with a disabled clustered index.

The *sp_updatestats* Stored Procedure

The *sp_updatestats* stored procedure runs `UPDATE STATISTICS` against all user-defined and internal tables in the current database. It specifies the `ALL` keyword against all user-defined and internal tables in the database. The stored procedure displays messages that indicate its progress. When the update is completed, it reports that statistics have been updated for all tables.

Evaluating Index Usage

A database index lets SQL Server quickly find specific information in a table or indexed view. It contains keys built from one or more columns in the table or view, and pointers that map to the storage location of the specified data. You can significantly improve the performance of database queries and applications by creating well-designed indexes to support your queries. Indexes can reduce the amount of data that must be read to return the query result set. Indexes can also enforce uniqueness on the rows in a table, ensuring the data integrity of the table data.

An index contains keys built from one or more columns in the table or view. These keys are stored in a structure known as a *b-tree* that enables SQL Server to find the row or rows associated with the key values quickly and efficiently. A table or view can contain clustered or nonclustered indexes.

Clustered Indexes

Clustered indexes sort and store the data rows in the table or view based on their key values. These are the columns included in the index definition. You can define only one clustered index per table because the data rows themselves can be sorted and stored in only one order.

Data rows in a table can be stored in order only when the table contains a clustered index. A table that has a clustered index is called a *clustered table*. If you do not define a clustered index for a table, SQL Server stores its data rows in an unordered structure called a *heap*.

Nonclustered Indexes

Nonclustered indexes have a structure separate from the data rows of the table itself. A nonclustered index contains nonclustered index key values and each key value entry has a pointer to the data row that contains the key value.

The pointer from an index row in a nonclustered index to a data row is called a *row locator*. The structure of the row locator depends on whether the actual data pages are stored in a heap or a clustered table. For a heap, a row locator is a pointer to the row. For a clustered table, the row locator is the clustered index key.

NOTE SQL Server 2005 enhancements

In SQL Server 2005, you can add nonkey columns to the *leaf level* of the nonclustered index to bypass existing index key limits (900 bytes and 16 key columns) and execute fully covered, indexed queries. A query is *covered* when the index contains all columns in the query.

Constraints

A constraint is a rule that the database server enforces automatically. Indexes are automatically created when you define PRIMARY KEY and UNIQUE constraints on table columns. For example, when you create a table and identify a particular column as the primary key, the SQL Server 2005 database engine automatically creates a PRIMARY KEY constraint and index on that column.

Query Optimizer

Well-designed indexes can reduce disk I/O operations and consume fewer system resources, thereby improving query performance. Indexes can be helpful for a variety of queries that contain SELECT, UPDATE, or DELETE statements. For example, suppose that you issue the following query against the AdventureWorks database:

```
SELECT Title, HireDate FROM HumanResources.Employee WHERE EmployeeID = 222
```

When this query is executed, the query optimizer evaluates each available method for retrieving the data and selects the most efficient method. The method used might be to scan a table or scan one or more indexes if they exist.

When performing a table scan, the query optimizer reads all the rows in the table and extracts the rows that meet the criteria of the query. A table scan generates many disk I/O operations and can be resource intensive. However, a table scan could be the most efficient method if, for example, the result set of the query is a high percentage of rows from the table. For example, if you execute a query that returns all rows in a table, SQL Server performs a table scan.

When the query optimizer uses an index, it searches the index key columns, finds the storage location of the rows needed by the query, and extracts the matching rows from that location. Generally, searching the index is much faster than searching the table because, unlike a table, an index typically contains few columns per row and the rows are in sorted order.

The query optimizer typically selects the most efficient method when executing queries. However, if no indexes are available, the query optimizer must use a table scan. SQL Server 2005 provides the DTA to help with the analysis of your database environment and in the selection of appropriate indexes.

Defragmenting Indexes

Data stored inside the database files can become fragmented at the index level. This prevents SQL Server 2005 from using indexes optimally. Index fragmentation can be either internal or external. Pages that have a lot of free space are internally fragmented. This

happens when you remove rows by DELETE statements or when SQL Server splits and only partly fills pages. Empty space on pages means there are less rows per page, which in turn means more page reads.

External fragmentation occurs when pages are not contiguous. Inserting new rows and updating existing rows can result in page splits. When a new page is created from a page split, SQL Server allocates it in the same 8-page extent as the original page if there is room for it. If the extent is already full, SQL Server allocates a new extent to the index or table and places the new page there. Thus, the new page is not contiguous to the original page.

You can use DBCC SHOWCONTIG to view the extent of index fragmentation. If you specify the TABLERESULTS option, you see extra output columns that describe index statistics. The following metrics assist you in auditing index defragmentation:

- **Avg. Page Density (full)** Shows how filled the pages are.
- **Scan Density** Shows the ratio between the Best Count of extents that should be necessary to read when scanning all the pages of the index and the Actual Count of extents that were read. This percentage should be as close to 100 as possible. Values less than 75 percent indicate serious external fragmentation.
- **Logical Scan Fragmentation** Shows the ratio of pages that are out of logical order. The percentage should be as close to 0 as possible, and any value over 10 percent indicates external fragmentation.

You can use the following methods to defragment an index:

- Drop and re-create the index
- CREATE INDEX ... WITH DROP_EXISTING
- DBCC DBREINDEX
- DBCC INDEXDEFRAG

You perform the first three options offline, which means that users cannot execute queries against the database while defragmentation is occurring. DBCC INDEXDEFRAG is an online operation, but SQL Server cannot defragment any indexes that are currently in use.

Rebuilding Indexes

You can use Object Explorer in SSMS to rebuild an index or all indexes on a table. Rebuilding an index drops and re-creates the index. This option removes fragmentation, reclaims disk space, and reorders the index rows in contiguous pages.

In Object Explorer, you connect to an instance of the SQL Server 2005 database engine, expand that instance, expand Databases and the relevant application database, navigate to the table that contains the index and then to the index itself. You then right-click the index, choose Rebuild, and click OK. To rebuild all the indexes in a table, you navigate to the table, right-click indexes, choose Rebuild All, and then click OK.

Creating Missing Indexes

When the query optimizer generates a query plan, it analyzes what the best indexes are for a particular filter condition. If the best indexes do not exist, the query optimizer generates a suboptimal query plan but stores information about the missing indexes. The missing indexes feature enables you to access information about these indexes so that you can decide whether you should implement them. You can then use missing index information to use CREATE INDEX statements that restore the missing indexes.

MORE INFO Missing indexes feature

For more information about this feature, search for “Using Missing Index Information to Write CREATE INDEX Statements” in Books Online or access [msdn2.microsoft.com/en-us/library/ms345405\(d=ide\).aspx](http://msdn2.microsoft.com/en-us/library/ms345405(d=ide).aspx).

Dropping an Inappropriate Index

When you no longer need an index, you can remove it from a database by using the DROP INDEX Transact-SQL statement or by connecting to the database with SSMS, navigating to the index in Object Explorer, right-clicking it, choosing Delete, and clicking OK. This process reclaims the disk space the index currently uses. Deleting an index is the same as dropping it.

MORE INFO Dropping indexes

For more information, search for “Dropping Indexes” in Books Online or access [msdn2.microsoft.com/en-us/library/ms190691\(d=ide\).aspx](http://msdn2.microsoft.com/en-us/library/ms190691(d=ide).aspx).

Dropping a clustered index can take time because in addition to dropping the clustered index, SQL Server must rebuild all nonclustered indexes on the table. To avoid rebuilding indexes, drop the nonclustered indexes first and the clustered index last.

Auditing and Analyzing Poorly Written Queries

Chapter 2 discusses query analysis in depth, so only a brief introduction is appropriate here. Your first step is to identify the queries you need to analyze. For example, you typically do not analyze queries that take less than (say) five seconds, although this time varies depending on your environment and the complexity of your queries. To prioritize query analysis, you should focus first on those queries that run the longest and the most frequently. If a query takes 60 seconds but runs only once per month, it is probably not a priority.

When troubleshooting query performance bottlenecks, you should first review the query itself. Review query code, looking for excessive data movement, unnecessary data sorting, cursors, long-running transactions, excessive data retrieval, and query syntax errors. Review the execution plan of the query to determine the individual segments of query code that perform poorly. Look for code segments that perform table or clustered index scans, bookmark lookups, index scans, data sorts, or improper data joins. Analyze the resources that are used by a query, such as CPU time, memory, and disk I/O.

The query environment includes many factors—for example, table design, column design, table indexes, column statistics, the number of applications executing on the server when the query is running, database configuration, server configuration, and the amount of data contained within the table. Any one of these environmental factors can affect the ability of a query to perform satisfactorily. Several tools are available to assist you in analyzing poorly written queries.

The Graphical Execution Plan

SSMS enables you to analyze a query plan and receive assistance to improve the query performance. The graphical execution plan uses icons to represent the execution of specific statements and queries in SQL Server, and it displays the data retrieval methods chosen by the SQL Server query optimizer. SSMS shows which statistics are missing and then helps you create those missing statistics.

To use the graphical execution plan, you access the Query Editor in SSMS and either open a Transact-SQL script that contains the queries you want to analyze or type the script directly into the Query Editor. After you load the script into the Query Editor, you can choose to display either an estimated execution plan or the actual execution

plan. If you click Display Estimated Execution Plan on the toolbar, SQL Server parses the script and generates an estimated execution plan. If you click Include Actual Execution Plan, you must execute the script before the execution plan is generated. After SQL Server parses or executes the script, you click the Execution Plan tab to see a graphical representation of execution plan output.

To view the execution plan, click the Execution plan tab in the results pane. Each query in the batch that is analyzed is displayed, and the analysis includes the cost of each query as a percentage of the total cost of the batch.

MORE INFO Execution plan

For more information, search for "execution plan" in Books Online or access msdn.microsoft.com/library/default.asp?url=/library/en-us/optimsq/odp_tun_1_5pde.asp.

Profiler

You can use SQL Server Profiler to capture a trace when a script containing the queries you want to analyze is running, or when you execute the queries directly by typing them into the Query Editor. Alternatively, you can capture a trace during a period of peak activity and filter the results to display the information you require. Profiler enables you to select the events you want to capture and specify the data columns from those events. You can use filters to obtain only the data you require. The events you should capture are the following:

- Stored Procedures—RPC:Completed
- TSQL—SQL:BatchCompleted

The first of these events captures stored procedures, and the second one captures all other Transact-SQL queries. Typically, the data columns you would specify are the following:

- Duration
- Event Class
- DatabaseID
- TextData
- CPU
- Writes
- Reads

- StartTime
- EndTime
- ApplicationName
- NTUserName
- LoginName
- SPID

StartTime, EndTime, ApplicationName, NTUserName, and LoginName are optional. You require DatabaseID only if you are accessing more than one database. Information in the Duration and TextData columns is especially useful in this analysis. Typically, you filter the trace to exclude system events and events with a duration of less than, for example, 5000 (5 seconds).

Database Engine Tuning Advisor

You can use the DTA to tune databases and create an optimized query environment. The tool enables you to view the effects of a workload against single or multiple databases, and it makes recommendations to alter the physical design structure of your SQL Server databases. These physical structures include clustered indexes, nonclustered indexes, indexed views, and partitioning. The goal of the DTA recommendations is to enable the query server to execute the query in the least amount of time. Typically, you run DTA on a test server because it uses a large amount of server resources.

You can use the DTA tool to troubleshoot issues caused by the existing indexing strategy and to perform a *what-if* analysis on the databases. A what-if analysis helps you to determine how an alternative indexing strategy might affect query performance without changing your current strategy.

Monitoring Transaction Log Size

Every SQL Server 2005 database has a transaction log that records all transactions and the database modifications made by each transaction. The transaction log is a critical component of the database and, in the case of a system failure, enables you to restore mission-critical data to the point of failure. You should never delete or move transaction logs unless you fully understand the consequences of doing so.

An application database is typically very active, with many transactions. This activity can cause the transaction log to grow quickly. The instance database transaction log

is also likely to grow rapidly. In both cases, you (or the database designer) should specify an initial log file size equal to 25 percent of the initial application database size. If the log is truncated regularly during log file backups or during a checkpoint, it should stay a reasonable size.

The transaction log is implemented as a separate file or set of files in the database. You can configure the files to expand automatically by setting the FILEGROWTH value for the log. This reduces the potential for running out of space in the transaction log, while at the same time reducing administrative overhead.

You can set % Free Space alerts that tell you when the physical or logical disk that holds the transaction files is running out of space, or you can use the SQLServer: Databases: Percent Log Used counter to trigger a SQL Server performance condition alert that starts a transaction log backup and sends you a message by using Database Mail. However, the best method of ensuring that transaction logs do not become too large is to implement a regular backup routine that truncates the logs.

The active portion of a transaction log cannot be truncated or removed by shrinking, and truncation can be delayed when log records remain active for a long time. This can happen for a variety of reasons—for example, no checkpoint has occurred since the last log truncation, a data backup or restore is in progress, a long-running transaction is active, database mirroring is paused, a database snapshot is being created, or a log scan is occurring.

SQL Server 2005 introduces the `log_reuse_wait` and `log_reuse_wait_desc` columns of the `sys.databases` catalog view. These features enable you to discover what is preventing log truncation.

MORE INFO `sys.databases` catalog view

For more information, search for "*sys.databases (Transact-SQL)*" in Books Online or access [msdn2.microsoft.com/en-us/library/ms178534\(SQL.90,d=ide\).aspx](http://msdn2.microsoft.com/en-us/library/ms178534(SQL.90,d=ide).aspx).

Monitoring Database Growth

When you create a database, you must either specify an initial size for the data and log files or accept the default size. As data is added to the database, these files become full. However, you must consider whether and how the database will grow beyond the initial space you allocate if more data is added to the database than will fit in the files.

By default, the data files grow as much as required until no disk space remains. Therefore, if you do not want the database files to grow any larger than when they were first created, you must specify this at the time you create the database using SSMS or the CREATE DATABASE statement.

Alternatively, SQL Server lets you create data files that can grow automatically when they fill with data, but only to a predefined maximum size. This can prevent the disk drives from running out of disk space completely. When you create a database, make the data files as large as possible, based on the maximum amount of data you expect in the database. Permit the data files to grow automatically, but put a limit on the growth by specifying a maximum data file growth size that leaves some available space on the hard disk. This approach lets the database grow if more data is added than expected, but it does not fill up the disk drive.

If the initial data file size is exceeded and the file starts to grow automatically, re-evaluate the expected maximum database size and add more disk capacity if required.

If the database is not supposed to expand beyond its initial size, set the maximum growth size of the database to zero. This prevents the database files from growing. If the database files fill with data, no more data is added until more data files are added to the database or until the existing files are expanded.

Filegroups

Filegroups are named collections of files used to simplify data placement and administrative tasks such as backup and restore operations. They use a proportional fill strategy across all the files within each filegroup. As data is written to the filegroup, the database engine writes an amount proportional to the free space in the file to each file within the filegroup, instead of writing all the data to the first file until full. For example, if file myfile1 has 300 MB free and file myfile2 has 600 MB free, one extent is allocated from file myfile1, two extents from file myfile2, and so on. Both files become full at about the same time, and the procedure achieves simple striping.

When all the files in a filegroup are full, the database engine automatically expands one file at a time in a round-robin manner to allow for more data (provided that the database is set to grow automatically). Suppose, for example, a filegroup is made up of three files. When space in all the files is exhausted, the first file is expanded. When the first file is full and no more data can be written to the filegroup, the second file is expanded. When the second file is full and no more data can be written to the

filegroup, the third file is expanded. If the third file becomes full and no more data can be written to the filegroup, the first file is expanded, and so on.

MORE INFO Physical database files and filegroups

For more information, search for “Physical Database Files and Filegroups” in Books Online, or access [msdn2.microsoft.com/en-us/library/ms179316\(d=ide\).aspx](http://msdn2.microsoft.com/en-us/library/ms179316(d=ide).aspx).

If you let files grow automatically and several files share the same disk, this can cause fragmentation. Therefore, you should create the files or filegroups on as many different local physical disks as possible. Also, put objects that compete heavily for space in different filegroups.

Improving Database Performance

Using files and filegroups improves database performance because it lets you create a database across multiple disks, multiple disk controllers, or RAID systems. For example, if your computer has four disks, you can create a database that is made up of three data files and one log file, with one file on each disk. As data is accessed, four read/write heads can access the data in parallel at the same time. This approach speeds up database operations.

Additionally, files and filegroups enable data placement because you can create a table in a specific filegroup. This improves performance because it enables you to direct all I/O for a specific table at a specific disk. For example, you can put a heavily used table on one file in one filegroup, located on one disk, and put the other less heavily accessed tables in the database on the other files in another filegroup, located on a second disk.

Monitoring Database Size

The *sp_spaceused* Transact-SQL stored procedure displays the number of rows, disk space reserved, and disk space used by a table, indexed view, or SQL Server 2005 Service Broker queue in the current database. It also displays the disk space reserved and used by the whole database. The following example summarizes space used in the AdventureWorks database and uses the optional parameter *@updateusage* to ensure current values are returned:

```
USE AdventureWorks;
GO
EXEC sp_spaceused @updateusage = N'TRUE';
GO
```

Investigating Locks and Deadlocks

The SQL Server database engine uses locks to synchronize access by multiple users to the same piece of data at the same time. Before a transaction reads or modifies data, it must protect itself from the effects of another transaction modifying the same data. It does this by requesting a lock on the piece of data. Locks have different modes, such as shared or exclusive. No transaction can be granted a lock that would conflict with the mode of a lock already granted on that data to another transaction. If this situation occurs, the database engine pauses the requesting transaction until the first lock is released.

When a transaction modifies a piece of data, it holds the lock protecting the modification until the end of the transaction. All locks held by a transaction are released when the transaction either commits or rolls back. Typically, applications do not request locks directly. Locks are managed internally by the database engine lock manager. When an instance of the database engine processes a Transact-SQL statement, the database engine query processor determines which resources are to be accessed. The query processor determines what types of locks are required to protect each resource based on the type of access and the transaction isolation level setting. The query processor then requests the appropriate locks from the lock manager. The lock manager grants the locks if there are no conflicting locks held by other transactions.

Displaying Locking Information

The database engine and its associated APIs provide mechanisms for displaying information about the locks currently held in a database. You can keep track of information about locks and lock notification requests by using the *sys.lock_information* DMV. The *sys.lock_information* is a virtual table that contains a collection of lock information—for example, the session that requested the lock, the resource that is being locked, the row identifier of a locked row within a table, the lock mode that is being requested or that has been granted, the internal identity of the table, and the status of the lock.

Deadlocks

A deadlock occurs when two SPIDs are waiting for a resource and neither process can advance because the other process is preventing it from getting the resource. When the lock manager's deadlock detection algorithm detects a deadlock, the lock manager chooses one of the SPIDs and kills the process. This frees the resources and allows the other SPID to continue. However, any application that depends on that process crashes.

By default, the lock manager chooses the transaction that is least expensive (for example, in time taken to load and initialize) to roll back as the deadlock victim. You can, however, specify the priority of sessions in a deadlock situation using the SET DEADLOCK_PRIORITY statement. You can set the DEADLOCK_PRIORITY to LOW, NORMAL (the default), or HIGH—or to any integer value in the range -10 through 10. If two sessions have different deadlock priorities, SQL Server chooses the session with the lower priority as the deadlock victim. If both sessions have the same deadlock priority, SQL Server chooses the session with the transaction that is least expensive to roll back. If sessions involved in the deadlock cycle have the same deadlock priority and the same cost, SQL Server chooses a victim at random.

The database engine provides three tools for viewing deadlock information:

- **Deadlock graph** An event group in SQL Server Profiler that presents a graphical description of the tasks and resources involved in a deadlock.
- **Trace flag 1204** This returns the type of locks participating in the deadlock and the current command affected. The results are captured in the SQL Server 2005 error log.
- **Trace flag 1222** This returns the type of locks participating in the deadlock and the current command affected in an XML-like format. The results are captured in the SQL Server 2005 error log.

MORE INFO **Deadlock graphs**

For more information, search Books Online for "Analyzing Deadlocks with SQL Server Profiler" or access [msdn2.microsoft.com/en-us/library/ms188246\(d=ide\).aspx](http://msdn2.microsoft.com/en-us/library/ms188246(d=ide).aspx).

Activity Monitor

You can use Activity Monitor within SSMS to get information about users' connections to the database engine and the locks that they hold. To open the tool within SSMS, connect to the server with Object Explorer, expand Management, and then double-click Activity Monitor.

Activity Monitor has three pages. The Process Info page contains information about the connections. The Locks By Process page sorts the locks by the connection. The Locks By Object page sorts the locks by the object name. You can click the Filter button to apply a filter and reduce the amount of information displayed. You can use Activity Monitor when troubleshooting database locking issues and to terminate a deadlocked or otherwise unresponsive process.

NOTE You can use a DMV.

You can use Activity Monitor to identify an unresponsive process. An alternative method is to access the `sys.dm_tran_locks` DMV, select from the `sys.syslockinfo` compatibility view, and locate the relevant SPID.

Optimizing RAID Configuration

SQL Server 2005 servers in a production environment typically use hard disk arrays configured to the appropriate RAID level. Typically, most organizations implement hardware RAID instead of software RAID. Hardware implementations are more expensive, but they are usually faster than software implementations and offer considerable advantages. Reliable disk storage that offers fast disaster recovery is relatively inexpensive when compared with the costs of data loss or long downtimes. Table 1-2 lists the advantages and disadvantages of using the various RAID implementations typically used with SQL Server 2005.

Table 1-2 RAID Implementations

RAID Implementation	Advantage	Disadvantage
Hardware-based RAID3 or RAID5	Excellent performance. Does not compete for processor cycles.	Cost
Hardware-based RAID1	Excellent redundancy. Does not compete for processor cycles.	Additional cost because of additional hardware
Hardware-based RAID10	Excellent performance. Excellent redundancy.	Additional cost because of additional hardware
Windows mirrored volumes (RAID1)	Good redundancy. Low cost.	Uses system processing resources
Windows RAID5	Excellent read performance. Low cost.	Uses system processing resources

RAID0, striping (with no parity), provides excellent read/write performance but no disaster recovery.

Filegroups, discussed earlier in this lesson, can provide advantages when configuring disk storage.

Troubleshooting Database and Transaction Log Storage

The SQLServer:Databases object provides counters to monitor bulk copy operations, backup and restore throughput, and transaction log activities. These counters enable you to monitor transactions and the transaction log to determine how much user activity is occurring in the database and how full the transaction log is becoming. Monitoring user activity and resource usage can help you to identify performance bottlenecks. You can monitor multiple instances of the object, each representing a single database, at the same time. The following list shows counters specific to monitoring database and transaction log storage:

- **Data File(s) Size (KB)** Returns the cumulative size (in kilobytes) of all the data files in the database, including any automatic growth
- **Log Cache Hit Ratio** Returns the percentage of log cache reads satisfied from the log cache
- **Log File(s) Size (KB)** Returns the cumulative size of all the transaction log files in the database
- **Log File(s) Used Size (KB)** Returns the cumulative used size of all the log files in the database
- **Percent Log Used** Returns the percentage of space in the log that is in use

NOTE Log Growths and Log Shrinks counters

The Log Growths and Log Shrinks counters return the total number of times the transaction log for the database has been expanded or shrunk. They do not indicate whether the log is running out of space.

You can use both SQL Server and Windows Server 2003 counters to troubleshoot disk space storage. For example, a SQL Server performance condition alert triggered by the SQLServer:Databases: Percent Log Used counter lets you know when a particular log file is running out of space. A Windows Server 2003 performance alert triggered by the LogicalDisk: % Free Space performance counter lets you know whether the RAID that holds the transaction logs or the database files is running short of disk space.

PRACTICE Using the Database Engine Tuning Advisor

In this practice, you use the DTA to analyze database performance against a workload generated by the trace you captured in Lesson 2, “Troubleshooting Connectivity to a SQL Server Instance,” by using Profiler. As with previous “Practice” sections in this chapter, the purpose of this practice session is to help you to become familiar with the tool rather than to perform detailed, in-depth analysis of a production system.

CAUTION A workload must contain tunable events.

DTA can analyze a workload file or table only if it contains tunable events—for example, SQL:BatchStarting, SQL:BatchCompleted, RPC:Starting, RPC:Completed, SP:StmtStarting, or SP:StmtCompleted. Unfortunately, none of the sample trace files supplied with the AdventureWorks database meet this criterion.

► Practice 1: Using the DTA to Analyze Database Performance

To use the DTA to analyze database performance against a workload generated by the trace you captured in Lesson 2 by using Profiler, follow these steps:

1. Log in to your domain at your member server using either your domain administrator account or an account that has been added to the sysadmin server role. If you are using virtual machine software, log in to your domain and connect to your member server.
2. From the All Programs menu, choose Microsoft SQL Server 2005, choose Performance Tools, and then choose Database Engine Tuning Advisor. Connect to your member server, specifying the AdventureWorks database and the TCP/IP network protocol on the Connection Properties tab.
3. On the General tab, specify the AdventureWorks database (unless it is already specified). Select the trace file you generated in Lesson 2 as the workload, as shown in Figure 1-16. Select Save Tuning Log.

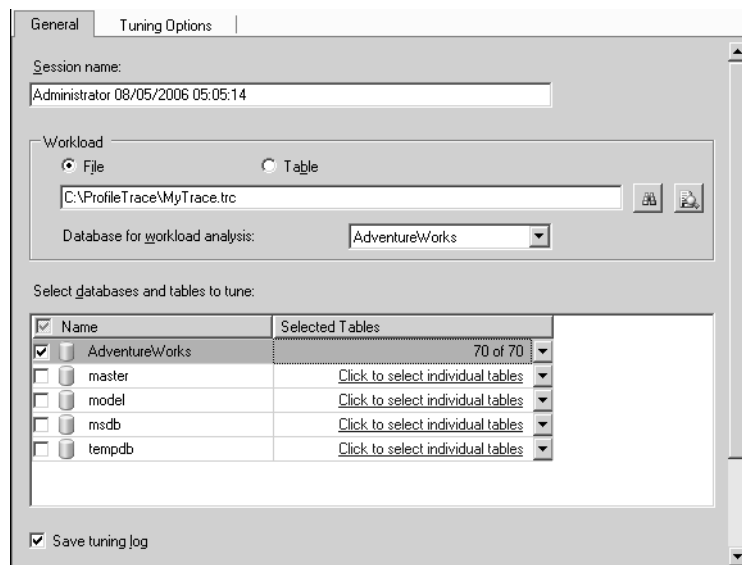


Figure 1-16 Specifying the database and workload.

4. Beside the AdventureWorks database, click the Click To Select Individual Tables link to list all the tables in the database. Verify that all tables are selected, as shown in Figure 1-17. (If they are not, select the check box next to Name to select all tables.) Do not change this default. Click anywhere on the tab to close the table list.

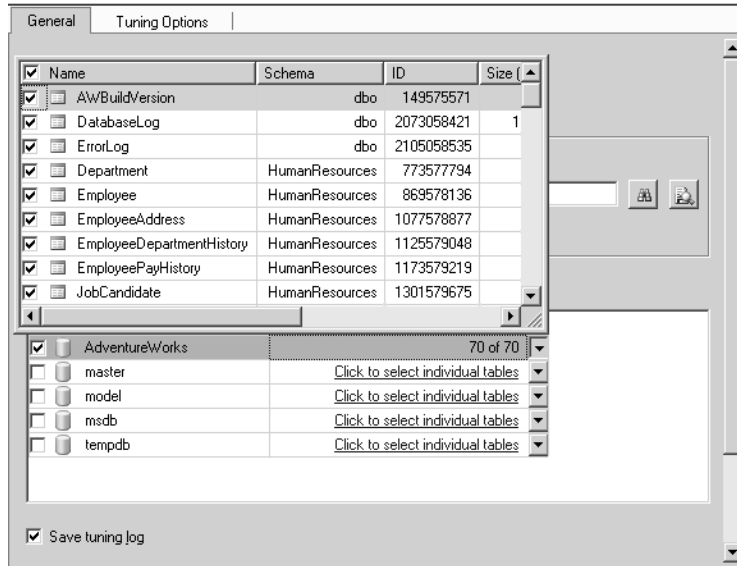


Figure 1-17 Selecting database tables.

5. Click the Tuning Options tab, and note the options available. Do not change any of them. Click Advanced Options, and note the advanced options. The facility to define a maximum space for recommendations and a maximum number of columns per index can be useful in a production environment, where the DTA can generate large files. Do not change any advanced options. Click OK to close the Advanced Options dialog box.
6. Click Start Analysis. The analysis proceeds. When it completes, the Progress tab indicates success, as shown in Figure 1-18.
7. View the Recommendations tab. It is unlikely that any possibilities for improvement result from this analysis.
8. View the Reports tab. You can access several reports, as shown in Figure 1-19.

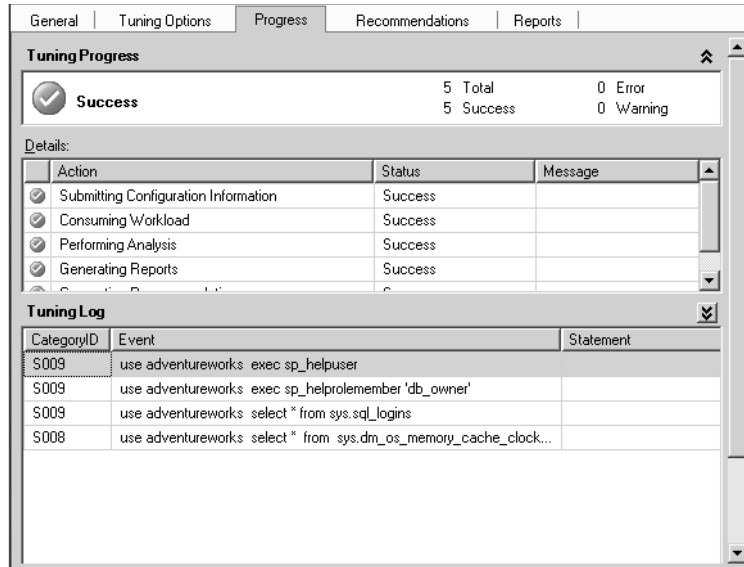


Figure 1-18 Analysis is complete.

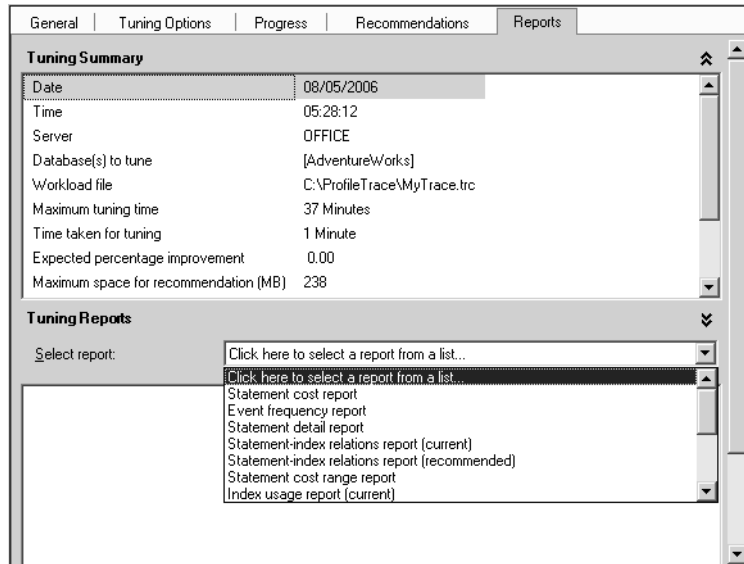


Figure 1-19 The Reports tab.

Lesson Summary

- You can use Profiler to monitor auto-grow and auto-shrink events in an instance of the database engine.
- You can use the stored procedure *sp_autostats* to display and change the auto-update statistics setting for a table, an index, or a statistics object.
- The query optimizer typically selects the most efficient method when executing queries. You can use Object Explorer in SQL Server Management Studio to manage indexes.
- You can use the graphical execution plan, Profiler, and the DTA to assist you in analyzing poorly written queries.
- The *sp_spaceused* Transact-SQL stored procedure displays the disk space reserved and used by the whole database, and also by such elements in the database as tables and indexed views.
- The Profiler deadlock graph event group presents a graphical description of the tasks and resources involved in a deadlock. You can use Activity Monitor to get information about users' connections to the database engine and the locks that they hold.
- You can use SQLServer:Databases object counters to enable you to monitor transactions, and you can use the transaction log to determine how much user activity is occurring in the database and how full the transaction log is becoming.

Lesson Review

You can use the following questions to test your knowledge of the information in Lesson 3, "Troubleshooting Database Performance." The questions are also available on the companion CD if you prefer to review them in electronic form.

NOTE Answers

Answers to these questions and explanations of why each answer choice is right or wrong are located in the "Answers" section at the end of the book.

1. You are a DBA with Contoso, Ltd. Users report that queries are taking a very long time to complete and database applications are crashing. You suspect deadlock errors. You need to find out the causes of the deadlocks. What should you do?
 - A. Run Profiler to create a trace with the Deadlock graph event group, and extract deadlock events
 - B. Use System Monitor to trace the application instance of the SQLServer-Locks: Number of Deadlocks/sec counter.
 - C. Use the *sys.dm_tran_locks* DMV.
 - D. Run the DTA and implement the recommendations.
2. The transaction log for an applications database is stored on a RAID10 volume. Auto-growth is not enabled. You suspect that the log might be running out of space. The storage capacity of the RAID10 array is not an issue. Which SQLServer:Databases counter should you monitor?
 - A. Log Cache Hit Ratio
 - B. Percent Log Used
 - C. Log Growths
 - D. Log Shrinks

Chapter Review

To further practice and reinforce the skills you learned in this chapter, you can perform the following tasks:

- Review the chapter summary.
- Review the list of key terms introduced in this chapter.
- Complete the case scenarios. These scenarios set up real-world situations involving the topics of this chapter and ask you to create a solution.
- Complete the suggested practices.
- Take a practice test.

Chapter Summary

- Windows Server 2003 Performance tools are used to determine whether memory, disk space, disk I/O, processor, or network resources are under stress on a SQL Server 2005 server.
- Pressure on the memory and processor resources can result from other major applications running on a SQL Server 2005 server. Stress on RAM memory can result in stress on disk I/O. Misconfigured server settings and badly written or wrongly indexed queries can cause memory stress.
- You can use Windows Server 2003 Performance Tools, Profiler, DTA, and DMVs to analyze and troubleshoot instance performance. A single tempdb database exists for each SQL Server instance, and tempdb issues can generate bottlenecks.
- You can configure the maximum number of user connections to a SQL Server instance. Microsoft recommends that you configure a SQL Server instance to use a dynamic port address by specifying a port address of 0 in the Server Network utility. You can control the maximum degree of CPU parallelism for individual query and index operation statements by specifying the MAXDOP query hint or MAXDOP index option.
- You can use the SQLServer:Wait Statistics and SQLServer:Latches performance objects to monitor waits, Profiler to monitor auto-grow and auto-shrink events in an instance of the database engine, the stored procedure *sp_autostats* to display and change the auto-update statistics settings, and Object Explorer in SQL Server Management Studio to manage indexes.

- You can use the graphical execution plan, Profiler, and the DTA to assist you in analyzing poorly written queries. The *sp_spaceused* Transact-SQL stored procedure displays the disk space reserved and used by a database and database elements.
- The Profiler deadlock graph event group presents a graphical description of the tasks and resources involved in a deadlock. You can use Activity Monitor to get information about users' connections to the database engine and the locks that they hold. You can use SQL Server: Databases object counters to enable you to monitor transactions and the transaction log.

Key Terms

Do you know what these key terms mean? You can check your answers by looking up the terms in the glossary at the end of the book.

- deadlock
- lock
- online analytical processing (OLAP)
- paging
- performance alert
- performance counter log
- performance trace log
- query
- small computer system interface (SCSI)
- source provider
- transaction log

Case Scenario

In the following case scenario, you will apply what you've learned in this chapter. You can find the answers to the questions in the "Answers" section at the end of this chapter.

Case Scenario: Resolving Physical Server and Database Bottlenecks

You are a DBA with Trey Research and are responsible for a SQL Server 2005 member server named SQLA that is located at company headquarters. SQLA contains the company's Human Resources (HR) database, customer database, and research projects database. Trey Research has expanded rapidly in the last year, resulting in an expansion of the number of employees, customers, and research projects. As a result, the response time for queries run against all three databases, but especially the research projects database, has increased noticeably.

Management has provided a budget to upgrade SQLA if necessary, but they want to ensure that expenditure is prioritized to resolve major system bottlenecks. You suspect that some queries are misconfigured, causing excessive recompiles. You are also concerned that transaction logs might be becoming full.

SQLA is running SQL Server 2005 SP1 Enterprise Edition on a Windows Server 2003 member server in the Trey Research Active Directory domain. It is an SMP server with four Pentium 4 CPUs running at 3.0 GHz, with 4 GB of RAM, and six physical disks at 60 GB apiece. Of the six physical disks, the first two form a hardware RAID1 array that holds the operating system, SQL Server 2005, and the transaction log. The remaining four disks are used in a hardware RAID10 configuration and contain the HR, customer, and research projects databases. The current size of the research projects database is 10 GB, and the database has been growing at an average rate of 5 percent per month for the past year. Customer growth is occurring at the same rate. The current size of the HR database is 5 GB, and this database has been growing at an average rate of 1.5 percent per month for the past year. At present, there is no failover server.

Peak activity for the database server occurs between 6 P.M. and 9 P.M. local time. You have been collecting counter logs during these times. Viewing these logs in System Monitor reveals the average values shown in Table 1-3.

Table 1-3 Performance Log Results for SQLA

Object: Counter	Average Reading
Processor: % Processor Time	60%
Memory: Pages/sec	25
Physical Disk: % Disk Time	42%

Table 1-3 Performance Log Results for SQLA

Object: Counter	Average Reading
Physical Disk: Avg. Disk Queue Length	2
SQL Server Buffer Manager: Buffer Cache Hit Ratio	89%

1. Which subsystem is the most likely cause of the long response times during hours of peak usage?
2. What short-term fix could improve SQL Server 2005 performance?
3. What tools can you use to analyze and optimize queries?
4. How can you detect excessive recompiles?
5. What graphical tool can you use to analyze deadlocks and find what is causing them?
6. You need to be alerted if a transaction log is becoming full. How should you configure this alert?

Suggested Practices

To help you successfully master the exam objectives presented in this chapter, complete the following tasks.

Troubleshoot Physical Server Performance

The main tools for troubleshooting physical server performance are the Windows Server 2003 (or Windows 2000 Server) Performance Tools, particularly Performance Logs and Alerts.

- **Practice 1: Use the Windows Server 2003 Performance Tools.** Configure counter logs, trace logs, and alerts. Investigate the use of instances. Click the Explain button to find out what the various counters record.

Troubleshoot Instance Performance

The main tool for troubleshooting instance performance is SQL Server Profiler.

- **Practice 1: Use SQL Server Profiler.** Use Profiler to generate and analyze traces generated by queries, stored procedures, and applications. Search the Internet for sample Profiler traces (for example, in online magazines).

Troubleshoot Database Performance

You can use several tools to troubleshoot database performance. These include SQL Server Management Studio, Transact-SQL statements, stored procedures, DMVs, SQL Server Profiler, and the DTA.

- **Practice 1: Use SQL Server Management Studio.** This tool is likely your main interface with SQL Server. Use Object Explorer, Query Editor, and Activity Monitor. Find out what additional features the tool provides.
- **Practice 2: Use Transact-SQL statements, stored procedures, and DMVs.** You are not expected to write large Transact-SQL applications, but you do need to be able to use Query Editor for administration and troubleshooting. Practice executing Transact-SQL statements, stored procedures, and DMVs from Query Editor. Build on the examples given in this chapter.
- **Practice 3: Use Profiler.** Generate Profiler traces, and investigate the Deadlock graph event group.
- **Practice 4: Use the DTA.** Run the DTA on your test network. Do not run this tool on a production server.

Take a Practice Test

The practice tests on this book's companion CD offer many options. For example, you can test yourself on just one exam objective, or you can test yourself on all the 70-444 certification exam content. You can set up the test so that it closely simulates the experience of taking a certification exam, or you can set it up in study mode so that you can look at the correct answers and explanations after you answer each question.

MORE INFO Practice tests

For details about all the practice test options available, see the "How to Use the Practice Tests" section in this book's Introduction.

Chapter 2

Analyzing Queries

You can often make database applications and stored procedures run faster on the same hardware by optimizing queries. Even ad hoc queries, run once to elicit specific information, can run faster if they follow the rules of query optimization. You can use such tools as SQL Server Profiler, the Database Engine Tuning Advisor (DTA), and dynamic management views (DMVs) to analyze a query and detect inefficiencies in query logic. You can add and optimize indexes—very often the key to efficient query performance. You can troubleshoot concurrency issues—for example, locks, blocks, deadlocks, and latches. However, often the best procedure is to display the queries in the Query Editor screen of SQL Server Management Studio (SSMS). With experience, you can gain insight into whether a query or procedure is efficiently constructed or badly formed. Hopefully, you can also educate your users in the basics of query construction.

Exam objectives in this chapter:

- Troubleshoot and maintain query performance.
 - Identify poorly performing queries.
 - Analyze a query plan to detect inefficiencies in query logic.
 - Maintain and optimize indexes.
 - Enforce appropriate stored procedure logging and output.
- Troubleshoot concurrency issues.

Lessons in this chapter:

- Lesson 1: Identifying Poorly Performing Queries 83
- Lesson 2: Analyzing a Query Plan to Detect Inefficiencies in Query Logic . . . 98
- Lesson 3: Maintaining and Optimizing Indexes 112
- Lesson 4: Enforcing Appropriate Stored Procedure Logging and Output . . . 132
- Lesson 5: Troubleshooting Concurrency Issues 137

Before You Begin

To complete the lessons in this chapter, you must have completed the following tasks:

- Configured a Microsoft Windows Server 2003 R2 computer with Microsoft SQL Server 2005 Enterprise Edition SP1 as detailed in the Appendix.
- Installed an updated copy of the AdventureWorks sample database as detailed in the Appendix.

No additional configuration is required for this chapter.

Real World

Ian Mclean

Often the analysis of queries is not particularly demanding. For example, I once did some work for a professional society with a large number of members worldwide, which frequently carried out searches based on the first few (typically three) letters of a member's surname. The Surname column stored data in mixed case by using case-sensitive collation. So that all members were located, the query used the UPPER function—for example:

```
SELECT Surname FROM Members WHERE UPPER(Surname) LIKE 'MAC%'
```

Every time such a query ran against the Members database, every record had the UPPER function applied to the Surname column. The queries were notoriously slow. The query needed to perform a case-insensitive search, and the collation could not be changed. My first step was to add a column to the database table:

```
ALTER TABLE Employees ADD UCSurname AS UPPER(Surname)
```

I could then create an index on the new column. As a result, the query could take the following form:

```
SELECT Surname FROM Members WHERE UCSurname LIKE 'MAC%'
```

Because an uppercase conversion was no longer required on every row every time such a query ran, time savings were significant.

Lesson 1: Identifying Poorly Performing Queries

This lesson discusses the methods and tools used to identify queries that are performing poorly, with unacceptably long execution times. Chapter 1, “Troubleshooting Database and Server Performance,” introduced tools you would typically use for this purpose:

- SQL Server Management Studio
- SQL Server Profiler
- DTA
- DMVs

This lesson discusses these tools. It also introduces the SQL Trace tool and discusses its use in query analysis.

NOTE Service packs

The service pack level at the time of this writing is Service Pack 1 (SP1). Unless otherwise indicated, all the information in the chapter applies to both SQL Server 2005 and SQL Server 2005 SP1.

After this lesson, you will be able to:

- Use Query Editor to list and analyze Transact-SQL queries and procedures.
- Use Profiler to identify problem queries.
- Use SQL Trace.
- Use the DTA to create an optimized query environment.
- Identify the DMVs that you can use to analyze query performance.

Estimated lesson time: 50 minutes

Using Query Editor

SQL Server Management Studio, introduced in Chapter 1, is your main interface with SQL Server 2005. You can use the Transact-SQL Query Editor in SSMS to view queries and procedures. The graphical execution plan in Query Editor lets you analyze queries in a batch and display the cost of each query as a percentage of the total cost of the batch.

Query Editor enables you to display, create, and run Transact-SQL scripts. You type scripts in the query window and execute them by pressing F5, by pressing Ctrl+E, or by clicking Execute on the toolbar or on the Query menu. If you highlight a portion

of the code in the window, only that portion is executed. The tool also provides help in analyzing and checking query syntax. For a simple syntax check, you can select a keyword in the Query Editor window and then either press F1 or click the Parse button on the toolbar. For more dynamic assistance, you can access the Help menu and select Dynamic Help. You can send a cancellation request to the server by clicking the Cancel Executing Query button on the toolbar. Some queries cannot be canceled immediately, but must wait for a suitable cancellation condition.

CAUTION Cancellation can take some time

In addition to the wait for a suitable cancellation condition, delays might occur while transactions are rolled back when you cancel a query. Do not expect cancellation to happen immediately.

You can click the Include Client Statistics button on the toolbar to open a Client Statistics window with statistics about the query, the network packets, and the elapsed time of the query. The Query Options button opens the Query Options dialog box. This dialog box lets you configure the default options for query execution and query results.

Switching to SQLCMD Mode

SQLCMD is a SQL command interpreter that runs against certain types of databases. The sqlcmd utility enables you to enter commands, including operating system commands, by using the Command Console. You can also execute SQLCMD commands from Query Editor by clicking the SQLCMD Mode button on the toolbar. You should precede SQLCMD commands with two exclamation points (!!) when using this facility.

If you select SQLCMD Mode, do not execute statements that prompt for a response. No interaction with the connection is possible to respond to the request, and the query continues to execute until you cancel it.

MORE INFO Editing SQLCMD scripts

For more information, search for "Editing SQLCMD Scripts with Query Editor" in Books Online, or access [msdn2.microsoft.com/en-us/library/ms174187\(d=ide\).aspx](https://msdn2.microsoft.com/en-us/library/ms174187(d=ide).aspx).

Viewing the Graphical Execution Plan

You can click Display Estimated Execution Plan on the Query toolbar to request a query execution plan from the query processor without executing the query, and display the

graphical execution plan in the execution plan window. An estimated plan parses the code in the Query Editor window and uses index statistics to estimate the number of rows SQL Server expects to return during each portion of the query execution.

If instead you click Include Actual Execution Plan on the toolbar, SQL Server executes the script before it generates the execution plan. After SQL Server parses or executes the script, you click the Execution Plan tab to see a graphical representation of execution plan output. The actual query plan SQL Server uses can differ from the estimated execution plan if the number of rows returned is significantly different from the estimate and the query processor changes the plan to be more efficient. Lesson 2 of this chapter discusses graphical execution plans in more detail.

The graphical execution plan uses icons to represent the execution of specific statements and queries in SQL Server, and it displays the data retrieval methods chosen by the SQL Server query optimizer. Each query in the batch that SQL Server analyzes is displayed, including the cost of each query as a percentage of the total cost of the batch. The Query Editor toolbar also lets you specify whether to display the query results as text or as one or more grids in the Results window, or to save the query results as a Report file with the .rpt extension.

Using Query Designer

You can open and edit an existing query in Query Designer by choosing Solution Explorer from the View menu while accessing the Query Editor pane or by clicking the Design Query In Editor button on the Query Editor toolbar. As shown in Figure 2-1, Query Designer has four panes: SQL, Criteria, Diagram, and Results. To open a query in all panes, choose Solution Explorer from the View menu, right-click the query you want to open, and click Open. To modify the query in Solution Explorer, highlight the SQL statements, right-click the highlighted area, and select Design Query In Editor. You cannot use Solution Explorer and Query Designer to analyze ad hoc queries. These tools analyze solutions or projects that database designers have implemented—for example, script projects. You can find examples of these by right-clicking the Solution Explorer pane, choosing Add, choosing New Project, and adding one of the three samples provided. You need to expand the project and click a query to activate the Design Query In Editor button.

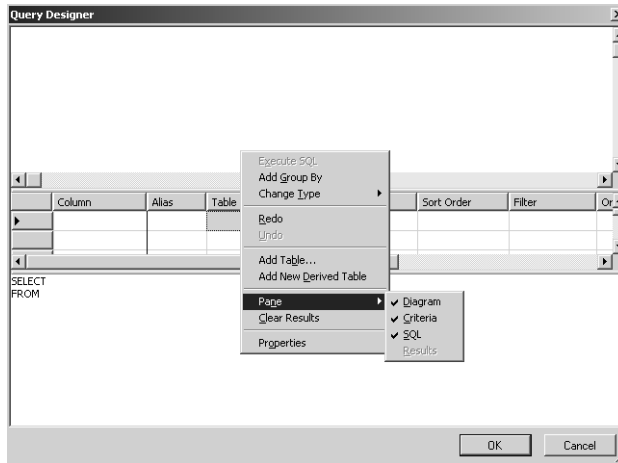


Figure 2-1 Query Editor panes.

Analyzing a Query in the Database Engine Tuning Advisor

In Chapter 1, you started SQL Server Profiler, captured a trace generated by running a query from the Query Editor, and then used that trace as a workload in the DTA. If you prefer, you can enter a query or procedure into Query Editor and use the Analyze Query In Database Engine Tuning Advisor button on the Query Editor toolbar to open the DTA and provide either all or a selected portion of the code in the Query Editor window as a workload for DTA analysis. We discuss the DTA later in this lesson.

Using SQL Server Profiler

SQL Server Profiler enables you to determine what Transact-SQL statements are submitted to a server. Typically, you run Profiler from a well-connected monitoring server to test your production server. The tool lets you create a trace that is based on a reusable template, watch the trace results as the trace runs, and store the trace results in a table. You can start, stop, pause, and modify the trace results as necessary and replay the trace results.

You can use SQL Server Profiler to capture a trace when a script containing the queries you want to analyze is running, or when you execute the queries directly by typing them into Query Editor. You can also capture a trace when an application is running and discover which queries in that application are executing slowly or using an excessive amount of Input/Output (I/O) or central processing unit (CPU) resources. Profiler lets you select the events you want to capture and specify the data columns from those events. You can use filters to obtain only the data you require.

You can capture all the Transact-SQL queries running against a database by tracing only two events. The Stored Procedures–RPC:Completed event captures stored procedures, and the TSQL–SQL:BatchCompleted event captures all other Transact-SQL queries. You can inspect the information captured in the relevant Profiler columns. For example, if you are looking for events that use a lot of CPU or I/O resources, you should look at the CPU, Writes, and Reads columns. If you want to discover which queries are taking a long time to execute, you should look at the Duration column. Typically, you also use the Duration reading to filter the information. You are seldom interested in queries with a very short duration.

The ApplicationName column tells you what application is running a set of queries. If you are monitoring activity over a period of time but are interested in the activity generated by specific database applications, you could filter on that column. The LoginName column is useful when you have users running ad hoc queries against a database and you want to identify users who are submitting inefficient queries that are affecting performance.

Using Profiler to Display Query Plan Information

You can add Showplan event classes to a trace definition so that Profiler gathers and displays query plan information in the trace. You can extract Showplan events from the trace by using the Events Extraction Settings tab when you are configuring the trace, by using the Extract SQL Server Events option on the Profiler File menu, or by right-clicking specific events and choosing Extract Event Data.

MORE INFO Showplan event classes

For more information about Showplan event classes, search for “Analyzing Queries with Showplan Results in SQL Server Profiler” in Books Online, or access [msdn2.microsoft.com/en-us/library/ms187941\(d=ide\).aspx](https://msdn2.microsoft.com/en-us/library/ms187941(d=ide).aspx). Lesson 2 of this chapter discusses Showplan in more detail.

Using the Database Tuning Advisor

You can use the DTA to analyze a workload provided by a profiler trace, or by a query or query script that you have entered in Query Editor. Typically, you run the tool on a test server that duplicates some of the information on your production server. You can use the DTA to identify inefficiently indexed queries and to identify appropriate changes. It lets you perform a *what-if* analysis to determine how an alternative indexing strategy might affect query performance.

The DTA enables you to view the effects of a workload against single or multiple databases and makes recommendations to alter the physical design structure of your SQL Server databases. These physical structures include clustered indexes, nonclustered indexes, indexed views, and partitioning. DTA recommendations help you redesign your query environment and reduce query execution times.

Using SQL Trace

You can choose not to create traces by using Profiler, but instead use Transact-SQL stored procedures from within your own applications to create traces manually on an instance of the SQL Server database engine. This approach allows you to write custom applications specific to the needs of your enterprise. SQL Trace gathers events that are instances of event classes listed in the trace definition. These events can be filtered out of the trace or queued for their destination. The destination can be a file or *SQL Server Management Objects (SMOs)*, which can use the trace information in applications that manage SQL Server.

An Event Source is any source that produces the trace event (for example, Transact-SQL batches) or SQL Server events (for example, deadlocks). If you include the event class in a trace definition, the event information is gathered by the trace after an event occurs. Predefined filters are applied, and the trace event information is passed to a queue. You can use the following Transact-SQL system stored procedures and functions to create and run traces and to trace an instance of the SQL Server database engine:

- ***sp_trace_create*** Creates a trace
- ***sp_trace_setevent*** Adds or removes event classes
- ***sp_trace_setfilter*** Applies a new or modified filter to a trace
- ***sp_trace_setstatus*** Starts, stops, and closes the trace
- ***sp_trace_generateevent*** Creates a user-defined event
- ***fn_trace_geteventinfo*** Returns information about events included in a trace
- ***fn_trace_getinfo*** Returns information about a specified trace or all existing traces
- ***fn_trace_getfilterinfo*** Returns information about filters applied to a trace

MORE INFO Trace events

For more information about trace events, search for “SQL Server Event Class Reference” in Books Online or access [msdn2.microsoft.com/en-us/library/ms175481\(d=ide\).aspx](https://msdn2.microsoft.com/en-us/library/ms175481(d=ide).aspx).

Using Transact-SQL system stored procedures creates a *server-side trace*. This guarantees that no events are lost as long as there is space on the disk and no write errors occur. If the disk becomes full or the disk fails, the SQL Server instance continues to run, but tracing stops.

Quick Check

- You are the DBA for a direct marketing company. You want to find out if any queries run against the company's Customer database are taking an excessive amount of time to complete. You also need to evaluate the effectiveness of current indexing in the database and identify changes that would be appropriate. What should you do?

Quick Check Answer

- Use Profiler or system stored procedures to create a trace of all Customer database activity over a predefined period—a typical day or a peak period within that day. Run the DTA using the trace as a workload. Filter the trace to view events that have a long duration.

Using DMVs

DMVs, introduced in SQL Server 2005, are increasingly used for a variety of tasks, one of which is to identify poorly performing queries. You can use Profiler and the DTA to debug SQL Server queries, but they need to run on a monitoring or test server because they degrade performance on a production server. Procedures that use DMVs, on the other hand, can run in a production environment. You can use (for example) the following DMVs to identify and analyze poorly performing queries:

- `sys.dm_exec_query_stats`
- `sys.dm_os_wait_stats`
- `sys.dm_db_index_physical_stats`
- `sys.dm_tran_locks`

The `sys.dm_exec_query_stats` DMV

This DMV contains a row for each query plan in SQL Server's query plan cache. When SQL Server removes a plan from the cache, it eliminates the corresponding row from the view. The DMV returns the following aggregate performance statistics for cached query plans:

- The number of times a query plan has been recompiled.

- A *handle* to the query plan. If the query was executed as part of a batch, this is a binary hash of the batch's text. If the query is part of a stored procedure, you can use this value, together with *statement_start_offset* and *statement_end_offset*, to retrieve the SQL text of the query by calling the *sys.dm_exec_sql_text* dynamic management function.
- The last time the query was executed.
- The number of times the query has been executed.
- The total amount of the following resources consumed by all invocations of the query plan as well as the least and greatest amount of CPU consumed by a single invocation of the query plan:
 - CPU
 - Physical Reads
 - Logical Writes
 - Logical Reads
 - CLR Time
 - Elapsed Time

NOTE Run the query twice

An initial query of *sys.dm_exec_query_stats* can produce inaccurate results if a workload is currently executing on the server. More accurate results can be determined by rerunning the query after the workload completes.

The *sys.dm_os_wait_stats* DMV

This DMV returns information about the waits encountered by threads that are in execution. You can use the view to diagnose performance issues with SQL Server and also with specific queries and batches. The DMV returns the following information:

- ***wait_type*** Name of the wait type.
- ***waiting_tasks_count*** Number of waits on this wait type. SQL Server increments this counter at the start of each wait.
- ***wait_time_ms*** Total wait time for this wait type in milliseconds. This time is inclusive of *signal_wait_time*.
- ***max_wait_time_ms*** Maximum wait time on this wait type.
- ***signal_wait_time*** Difference between the time the waiting thread was signaled and the time it started running.

The wait type can be one of the following:

- **Resource wait** Resource waits occur when a thread (or worker) requests access to a resource that is not available because the resource is being used by another worker or is not yet available for some other reason. Examples of resource waits are locks, latches, network waits, and disk I/O waits. Lock and latch waits are waits on synchronization objects.
- **Queue wait** Queue waits occur when a worker is idle, waiting for work to be assigned. Queue waits are most typically associated with system background tasks such as the deadlock monitor and deleted record cleanup tasks. These tasks wait for work requests to be placed into a work queue. Queue waits might also periodically become active even when no new packets have been put on the queue.
- **External wait** External waits occur when a SQL Server worker is waiting for an external event to finish. When you diagnose blocking issues, you need to remember that external waits do not always imply that the worker is idle because the worker might be actively running external code.

Specific types of wait times during query execution can indicate bottlenecks within the query. Similarly, high wait times or server-wide wait counts can indicate bottlenecks in query interactions within the server instance. For example, lock waits indicate data contention by queries, page I/O latch waits indicate slow I/O response times, and page latch update waits indicate incorrect file layout.

All data returned by this DMV is cumulative since the last time you reset the statistics or started the server. Therefore, if the DMV returns a high value for (say) *max_wait_time_ms*, you need to determine whether this is a result of current performance problems. If it is, you need to reset the statistics in the DMV and query the counters again when new statistics have accumulated. You can reset the contents of this DMV view by running the following command:

```
DBCC SQLPERF ('sys.dm_os_wait_stats', CLEAR);GO
```

The *sys.dm_db_index_physical_stats* DMV

This DMV returns size and *fragmentation* information for the data and indexes of the specified table or view. Fragmentation occurs through the process of data modifications (INSERT, UPDATE, and DELETE statements) that are made against a table in a database and therefore to the indexes defined on the table. For queries that scan part or all of the indexes of a table, this fragmentation can cause additional page reads. The

fragmentation level of an index or *heap* (an unordered structure that stores data rows when a table has no clustered index) is shown in the `avg_fragmentation_in_percent` column returned by the DMV. For heaps, the value represents the extent fragmentation of the heap. For indexes, the value represents the logical fragmentation of the index.

NOTE Extent fragmentation

Extent fragmentation is the percentage of out-of-order extents in the leaf pages of a heap. An out-of-order extent is one for which the extent that contains the current page for a heap is not contiguous with the extent that contains the previous page.

For optimum performance, the value returned for `avg_fragmentation_in_percent` should be as close to zero as possible. Typically, however, values up to 10 percent are acceptable. You can use all methods of reducing fragmentation—such as rebuilding, reorganizing, or re-creating indexes—to reduce these values.

The *sys.dm_tran_locks* DMV

Locks are the normal mechanism by which threads and processes share resources. However, problems can occur when two or more tasks permanently block each other by each task having a lock on a resource that the other tasks are trying to lock. This can happen, for example, when two users issue conflicting ad hoc queries.

The *sys.dm_tran_locks* DMV returns information about currently active lock manager resources. Each row represents a currently active request to the lock manager for a lock that has been granted or is waiting to be granted. The columns in the result set are divided into two main groups—resource and request. The resource group describes the resource on which the lock request is being made, and the request group describes the lock request.

When you identify a lock contention, you can use the *sp_who* stored procedure to get information about the current users and processes that are involved. Alternatively, you can use Activity Monitor in SSMS to get information about user connections and the locks that they hold.

MORE INFO Additional DMVs

The DMVs described in this lesson are only a small subset of the views that are available to you. For more information, search for “SQL Server Operating System Related Dynamic Management Views” in Books Online or access [msdn2.microsoft.com/en-us/library/ms176083\(SQL.90,d=ide\).aspx](https://msdn2.microsoft.com/en-us/library/ms176083(SQL.90,d=ide).aspx).

PRACTICE Identifying a Badly Performing Query

In the following practice session, you run an inefficient query against a table in the AdventureWorks database and use Profiler to create a trace. You can extend the practice by using the DTA to analyze the workload.

► Practice: Using Profiler to Identify a Badly Performing Query

To generate a Profiler trace and capture a poorly performing query, complete the following steps:

1. Log in to your domain at your member server by using either your domain administrator account or an account that has been added to the sysadmin server role. If you are using virtual machine software, log in to your domain and connect to your member server.
2. Unless you have already done so in the Chapter 1 Practices, create a folder on your server to store Profiler traces—for example, C:\ProfileTrace.
3. From the Programs (or All Programs) menu, choose Microsoft SQL Server 2005, choose Performance Tools, and then choose SQL Server Profiler. On the File menu, choose New Trace. In the Connect to Server dialog box, specify your member server and click Options. On the Connection Properties tab, specify the AdventureWorks database and TCP/IP as the network protocol. Click Connect.
4. Give the trace a name—for example, **SlowQueryTrace**.
5. On the Events Selection tab of the Trace Properties dialog box, clear all check boxes except those for the ExistingConnection and TSQL—SQL:BatchCompleted events, as shown in Figure 2-2. Click Run.

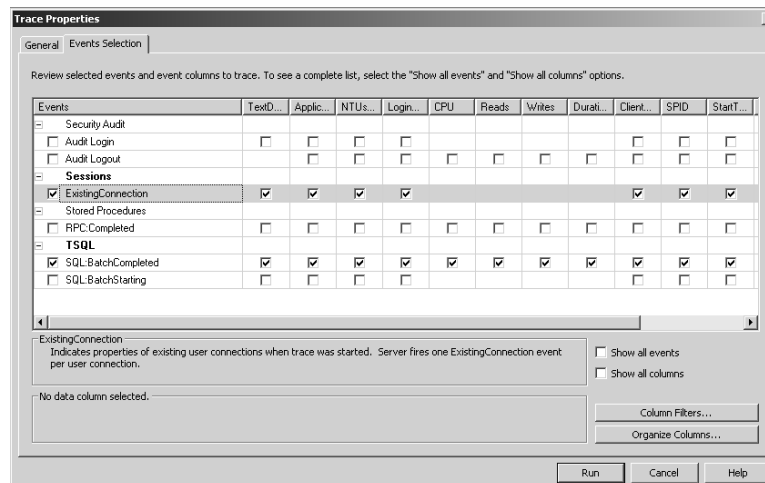


Figure 2-2 Profiler events selection.

6. From the Programs menu, select Microsoft SQL Server 2005 and click SQL Server Management Studio. Connect to the database engine on your member server. Specify Windows Authentication, the TCP/IP protocol, and the AdventureWorks database.
7. Click New Query to start the Query Editor
8. In the Query pane, type the following text:


```
USE AdventureWorks
GO
SELECT DISTINCT *
FROM HumanResources.Employee
WHERE SUBSTRING(NationalIDNumber,1,1) = '8'
```
9. Click the Parse button (indicated by a blue check mark). Your screen should be similar to Figure 2-3. Press F5 to run the query.

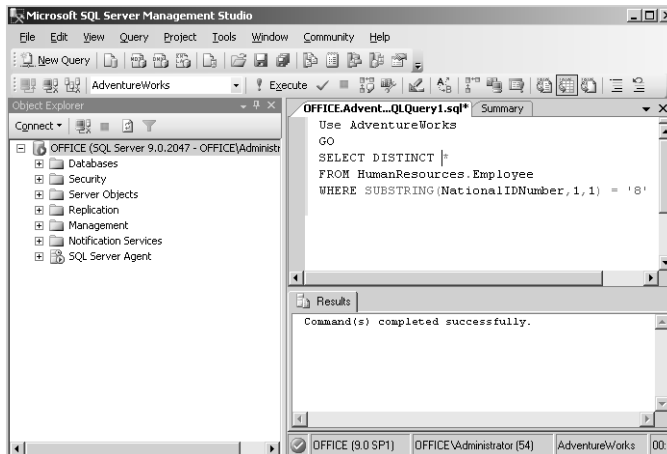


Figure 2-3 Parsing the query.

10. In Profiler, on the File menu, stop the trace, and then save it as a trace file in the folder you created.
11. You can use Profiler to view the trace. Figures 2-4 and 2-5 show some of the columns in the trace. High disk I/O usage (in this example) is indicated by a high number in the Reads column. A high number in the Writes column also indicates high disk I/O usage.

EventClass	NTUserName	LoginName	CPU	Reads
SQL:BatchCompleted	Adminis...	OFFICE...	110	419
SQL:BatchCompleted	Adminis...	OFFICE...	0	0
SQL:BatchCompleted	Adminis...	OFFICE...	30	169
SQL:BatchCompleted	Adminis...	OFFICE...	0	0

Figure 2-4 High CPU and disk I/O usage.

Duration	ClientProces...	SPID	TextData
0	2928	54	Use Adventureworks
13055	2928	54	SELECT * FROM HumanResources.E...
0	2928	54	SET PARSEONLY ON
0	2928	54	Use Adventureworks

Figure 2-5 Long duration.

- Repeat the preceding procedure to capture a trace for the following query in a new trace file:

```
USE AdventureWorks
GO
SELECT*
FROM HumanResources.Employee
WHERE NationalIDNumber LIKE '8%'
```

Compare the CPU, Reads, and Duration statistics, shown in Figures 2-6 and 2-7, with those shown in Figures 2-4 and 2-5. What are the possible causes of the differences in performance of these two queries?

EventClass	TextData	CPU	Reads
SQL:BatchCom...	SELECT * FROM HumanResource...	0	0
SQL:BatchCom...	SET PARSEONLY OFF	0	0
SQL:BatchCom...	Use AdventureWorks	0	0
SQL:BatchCom...	SELECT * FROM HumanResource...	10	15

Trace is stopped. Ln 41, Col 1 Rows: 41

Figure 2-6 Moderate CPU and disk I/O usage.

TextData	Duration	ClientProcessID	SPID
SELECT * FROM Hum...	11	2928	54
SET PARSEONLY OFF	0	2928	54
Use AdventureWorks	0	2928	54
SELECT * FROM Hum...	86	2928	54

Trace is stopped. Ln 41, Col 1 Rows: 41

Figure 2-7 Moderate duration.

By omitting the DISTINCT operator and not performing substring operations, the second query uses significantly less CPU and disk I/O resource than the first.

Lesson Summary

- You can use SQL Server Management Studio, SQL Server Profiler, the DTA, and DMVs to identify poorly performing queries.
- Query Editor enables you to check the syntax of a query plan and to access the graphical execution plan facility.

- You can add Showplan event classes to a trace definition so that Profiler gathers and displays query plan information in the trace.
- You can use Transact-SQL stored procedures from within your own applications to create traces manually on an instance of the SQL Server database engine.

Lesson Review

You can use the following questions to test your knowledge of the information in Lesson 1, “Identify Poorly Performing Queries.” The questions are also available on the companion CD if you prefer to review them in electronic form.

NOTE Answers

Answers to these questions and explanations of why each answer choice is correct or incorrect are located in the “Answers” section at the end of the book.

1. You are the DBA for a direct marketing organization. Your company uses a SQL Server 2005 application that enables users to run both predefined and ad hoc queries against a customer database. You suspect that some of these queries consume an excessive amount of server resources. You need to identify which queries consume the most resources. You want to achieve this goal as quickly as possible. What should you do?
 - A. Use the *sys.dm_exec_query_stats* DMV.
 - B. Use the *sys.dm_db_index_physical_stats* DMV.
 - C. Use Profiler, and include the Showplan event classes.
 - D. Use the *sqlcmd* utility.
2. You are investigating locking contention on a SQL Server 2005 server that has been in operation for several months. The *sys.dm_os_wait_stats* DMV is showing a high value in the *max_wait_time_ms* column. You need to find out whether this value is a factor in the current performance problems. You must also minimize the impact on database users. What should you do?
 - A. Restart the SQL Server computer.
 - B. Restart the SQL Server service.
 - C. Reset the statistics in the DMV.
 - D. Inspect the value in the *waiting_tasks_count* counter.

Lesson 2: Analyzing a Query Plan to Detect Inefficiencies in Query Logic

You can examine the query execution plan to discover why a query runs slowly and determine what is causing the problem. The query execution plan indicates how the SQL Server database engine navigates tables and uses indexes to access or process the query data. You can display execution plans by using the following methods:

- Transact-SQL SET statement options
- SQL Server Profiler event classes
- SQL Server Management Studio graphical execution plan

If you do not want to display the query execution plan in text, graphical, or extensible markup language (XML) mode, but instead want to obtain statistics about disk I/O operations or CPU usage, you can use DMVs—for example, *sys.dm_exec_query_stats*.

After this lesson, you will be able to:

- Analyze a query plan to detect excessive I/O activity.
- Analyze a query plan to detect an excessive number of table scans.
- Analyze a query plan to detect excessive CPU usage.

Estimated lesson time: 35 minutes

Detecting Excessive I/O Activity

When query execution takes significantly longer than expected, you need to first identify the slow-running queries and then determine the reason for the delay. Slow-running queries can be caused by performance problems related to your network, to the server on which SQL Server is running, or to the physical database design.

Sometimes the problem can be linked to indexes and statistics—for example, a lack of useful statistics on indexed columns, out-of-date statistics on indexed columns, a lack of useful indexes, or a lack of useful indexed views. Lesson 3 discusses indexes. Inadequate memory available for SQL Server can cause an excessive number of disk I/O operations. Several methods exist for determining whether a query execution plan results in excessive I/O activity. You can use the Transact-SQL SET statement options, Profiler event classes, the Management Studio graphical execution plan, or the *sys.dm_exec_query_stats* DMV.

Using the Transact-SQL SET Statement Options

After you have identified the slow-running query by using Profiler, SSMS, or a DMV, you can analyze the query's performance by using the Transact-SQL SET statement to enable the SHOWPLAN, STATISTICS IO, STATISTICS TIME, and STATISTICS PROFILE options:

- **SET SHOWPLAN_XML ON** Retrieves data in the form of a well-defined XML document
- **SET SHOWPLAN_TEXT ON** Returns execution information for each Transact-SQL statement without executing it
- **SET STATISTICS XML ON** Displays a result set after each executed query representing a profile of the execution of the query in the form of a set of well-defined XML documents
- **SET STATISTICS IO ON** Reports information about the number of scans, logical reads (pages accessed in cache), and physical reads (number of times the disk was accessed) for each table referenced in the statement
- **SET STATISTICS TIME ON** Displays the amount of time (in milliseconds) required to parse, compile, and execute a query
- **SET STATISTICS PROFILE ON** Displays a result set after each executed query representing a profile of the execution of the query

The SHOWPLAN option enables you to analyze query execution plans without needing to execute them, and it corresponds to the Showplan event classes in Profiler. SET SHOWPLAN_XML is the recommended option in SQL Server 2005 because it produces more detailed information than the SHOWPLAN_ALL and SHOWPLAN_TEXT SET options.

When you issue the SET STATISTICS IO ON statement before executing a query, SQL Server displays statistical information when the query runs. SQL Server returns this statistical information for all subsequent Transact-SQL statements until you set this option to OFF. You set STATISTICS IO ON at execute or run time and not at parse time. The option provides the following I/O-related statistics:

- **Table** Name of the table
- **Scan count** Number of scans performed
- **logical reads** Number of pages read from the data cache
- **physical reads** Number of pages read from disk
- **read-ahead reads** Number of pages placed into the cache for the query

The option also provides the following statistics for large object blocks (LOBs):

- **lob logical reads** Number of text, ntext, image, or large value type—varchar(max), nvarchar(max), varbinary(max)—pages read from the data cache
- **lob physical reads** Number of text, ntext, image, or large value type pages read from disk
- **lob read-ahead reads** Number of text, ntext, image, or large value type pages placed into the cache

NOTE LOB columns

When Transact-SQL statements retrieve data from LOB columns, some LOB retrieval operations might require traversing the LOB tree multiple times. This might cause SET STATISTICS IO to report higher-than-expected logical reads.

The following example shows how many logical and physical reads SQL Server uses as it executes a query against the AdventureWorks database:

```
USE AdventureWorks;
GO
SET STATISTICS IO ON;
GO
SELECT *
FROM Production.ProductCostHistory
WHERE StandardCost < 500.00;
GO
SET STATISTICS IO OFF;
GO
```

This example returns the following result set, which you can view on the Messages tab:

```
Table 'ProductCostHistory'. Scan count 1, logical reads 5,
physical reads 0, read-ahead reads 0, lob logical reads 0,
lob physical reads 0, lob read-ahead reads 0.
```

Using SQL Server Profiler Event Classes

You can select SQL Server Profiler event classes to include in traces that produce estimated and actual execution plans in XML or text. To display execution plan information by using these event classes, you must also include the appropriate Transact-SQL statements to turn on—for example, STATISTICS IO or SHOWPLAN_XML. The following event classes are used in SQL Server 2005.

- **Showplan XML** Captures the estimated execution plan in XML format with full compile-time details in the TextData data column of the trace.

- **Showplan XML For Query Compile** The compile time counterpart of the Showplan XML event. Showplan XML occurs when a query is executed. Showplan XML For Query Compile occurs when a query is compiled.
- **Showplan XML Statistics Profile** Captures the actual execution plan in XML format with full run-time details in the TextData data column of the trace.
- **Performance statistics** Occurs when a compiled query plan is cached for the first time, when it is compiled or recompiled any number of times, and when the plan is flushed from the cache. In some cases, the TextData data column for this event contains the plan in XML format that is being compiled or recompiled.

NOTE Event classes scheduled for deprecation

Microsoft recommends that you do not use the Profiler event classes: Showplan All, Showplan All For Query Compile, Showplan Statistics Profile, Showplan Text, and Showplan Text (Unencoded). In a future version of SQL Server, these event classes will be deprecated.

Using SQL Server Management Studio Graphical Execution Plan

The SSMS graphical execution plan is an interactive graphical tool that enables you to write queries, execute multiple queries simultaneously, view results, analyze the query plan, and receive assistance to improve the query performance. The execution plan options graphically display the data retrieval methods chosen by the SQL Server query optimizer. The graphical execution plan uses icons to represent the execution of specific statements and queries in SQL Server rather than the tabular representation produced by the Transact-SQL statement options. The graphical display is useful for understanding the performance characteristics of a query. However, you need to use this facility in conjunction with the SET STATISTICS IO ON Transact-SQL statement or the *sys.dm_exec_query_stats* DMV because it does not display actual disk I/O statistics.

NOTE Reading graphical execution plans

Reading graphical execution plans is something that you learn only through practice. You need to read them from right to left, and they can use a large number of different icons. You can obtain a list of the icons and a description of each by searching for "Graphical Execution Plan Icons (SQL Server Management Studio)" in Books Online or accessing [msdn2.microsoft.com/en-us/library/ms175913\(d=ide\).aspx](https://msdn2.microsoft.com/en-us/library/ms175913(d=ide).aspx). Study the descriptions carefully, and ensure that you can distinguish between (for example) a clustered index scan, a clustered index seek, and a table scan.

Using the *sys.dm_exec_query_stats* DMV

Lesson 1 discussed how you use this DMV to identify a poorly performing query. You can also use it to display I/O statistics. The disadvantage of this method is that you need to run the query (typically twice) to obtain statistics. You cannot display an estimated execution plan. The DMV returns statistics about a query execution plan, for example:

- The total number of physical reads performed by executions of this plan since it was compiled
- The number of physical reads performed the last time the plan was executed
- The minimum and maximum number of physical reads that this plan has ever performed during a single execution
- The total number of logical writes performed by executions of this plan since it was compiled
- The number of logical writes performed the last time the plan was executed
- The minimum and the maximum number of logical writes that the plan has ever performed during a single execution
- The total number of logical reads performed by executions of this plan since it was compiled
- The number of logical reads performed the last time the plan was executed
- The minimum and maximum number of logical reads that this plan has ever performed during a single execution

Monitoring Table Scans

Table scans are typically time consuming and can slow down query execution. Where the execution plan requires a table scan, query optimizer typically configures the scan as efficiently as possible. Excessive table scanning is caused by poorly designed queries, and you need to inspect and amend such queries in Query Editor. You should then work at eliminating unnecessary scans and redesigning scans that apply an operator to each row as the table is being scanned or scans that return more rows than are needed.

NOTE Table scans and indexes

A table scan can also occur when the table does not have an index that satisfies the query. However, it can be argued that this situation is caused by poor query design because required indexes do not exist or because queries are not designed to take advantage of existing indexes.

Earlier in this lesson, we discussed methods of obtaining disk I/O statistics. An excessive number of logical reads or writes (or both) can identify a plan that is causing excessive table scans. For example, the query might be carrying out a table scan instead of using an index. In this case, you need to review your indexes and create new indexes as required. Missing indexes can force table scans and slow down the query. Another approach is to use Profiler to create a trace of current activity and identify the queries with the longest durations. In general, a query execution plan that is taking a long time is likely due to an excessive number of table scans.

Limiting the Information Returned by a Table Scan

If a query returns unnecessary data, this wastes resources by causing SQL Server to perform additional I/O operations. In addition, it increases *network* traffic. A table scan places a shared lock on the table during the scan. If the table is large, the duration of the shared lock might prevent other users from modifying the table's data and thus hurt concurrency. If a query includes (for example) a SELECT statement but does not need to return every row that satisfies that statement, you should use a WHERE clause to limit the number of rows returned. The following example lists all rows, and every column within each row, of the Production.ProductCostHistory table:

```
USE AdventureWorks
GO
SELECT *
FROM Production.ProductCostHistory
```

You can restrict the output of this query with a WHERE clause—for example:

```
USE AdventureWorks
GO
SELECT * FROM Production.ProductCostHistory
WHERE StandardCost < 500.00;
```

If the query does not need to return the value in every column, you can save execution time by specifying the columns required—for example:

```
USE AdventureWorks
GO
SELECT ProductName, StandardCost
FROM Production.ProductCostHistory
WHERE StandardCost < 500.00;
```

You also need to create an index on the column you are using to specify the data you require. Even if you design a SQL Server query to avoid full table scans by using a WHERE clause to restrict the result set, it might not perform as expected unless you have an appropriate index supporting the query.

If an application allows users to run queries that could potentially return a large number of unnecessary rows, you should consider using the TOP operator in the SELECT statement. This way, you can limit how many rows SQL Server returns, even if the user does not enter any criteria to help reduce this number. However, you need to use this feature carefully. If you limit the number of rows returned to the first 100 and the information a user needs happens to be in the 101st row, you might not be the most popular DBA on the planet. Bearing this in mind, the following query limits the number of rows returned to 100:

```
USE AdventureWorks
GO
SELECT TOP 100 LoginID
FROM HumanResources.Employee
```

NOTE SET ROWCOUNT

SQL Server 2005 retains the SET ROWCOUNT statement. The SET ROWCOUNT value overrides the SELECT statement TOP keyword if the ROWCOUNT is the smaller value. Some instances exist (for example, rows returned from an unsorted heap) where the TOP operator is more efficient than using SET ROWCOUNT. Because of this, Microsoft recommends using the TOP operator rather than SET ROWCOUNT to limit the number of rows returned by a query.

You can also use the TOP operator to specify a percentage value. SQL Server 2005 introduces the WITH TIES argument, which enables you to return more than the specified number or percent of rows if the values of the last group of rows are identical—for example:

```
USE AdventureWorks
GO
SELECT TOP(10) PERCENT WITH TIES
LoginID, NationalIDNumber, Title, HireDate
FROM HumanResources.Employee ORDER BY HireDate DESC
```

NOTE Additional TOP operator enhancements

SQL Server 2005 enables you to use the TOP statement with data manipulation language (DML) statements—for example, DELETE, INSERT, and UPDATE. Also, you cannot use the TOP statement in conjunction with UPDATE and DELETE statements on partitioned views.

Tuning Queries

Tuning queries to reduce execution times is something you learn from experience. Reducing the number of rows and columns returned and ensuring that you have created indexes on the relevant columns are obvious steps. Some of the techniques I

explore in this section are not obvious, and the list is most definitely not exclusive. If I miss a tip or trick that you have found especially effective, or of which you are particularly fond, I apologize.

Using the DISTINCT Statement Using the DISTINCT statement in a SELECT query eliminates duplicate rows. At first sight, this seems like a good thing, and indeed it is if your tables have duplicate rows and having duplicates in the result set causes problems. However, the DISTINCT statement uses a lot of SQL Server resources and reduces the physical resources that other *SQL statements* have at their disposal. You need to carefully evaluate whether your SELECT query needs the DISTINCT clause. Unfortunately, some application developers automatically add this clause to every one of their SELECT statements and to their ad hoc queries.

Using the UNION Statement The UNION statement performs the equivalent action of SELECT DISTINCT on the final result set to eliminate any duplicate rows. This process occurs even when no duplicate records exist in the final result set. If you know that there are duplicate records and this presents a problem, you should use the UNION statement to eliminate the duplicate rows.

If you know that no duplicate rows exist, or if duplicate rows do not present a problem, you should use the UNION ALL statement instead of the UNION statement. UNION ALL does not perform the SELECT DISTINCT function, and it does not tie up SQL Server resources unnecessarily. Sometimes you might want to merge two or more sets of data resulting from two or more queries using UNION—for example:

```
USE AdventureWorks
GO
SELECT NationalIDNumber, Title
FROM HumanResources.Employee
WHERE NationalIDNumber LIKE '90%'
UNION
SELECT NationalIDNumber, Title
FROM HumanResources.Employee
WHERE Title = 'Marketing Manager'
```

This query runs faster if you rewrite it as follows:

```
USE AdventureWorks
GO
SELECT DISTINCT NationalIDNumber, Title
FROM HumanResources.Employee
WHERE NationalIDNumber
LIKE '90%' OR Title = 'Marketing Manager'
```

If neither criterion is likely to return duplicate rows, you can further boost query performance by removing the `DISTINCT` clause:

```
USE AdventureWorks
GO
SELECT NationalIDNumber, Title
FROM HumanResources.Employee
WHERE NationalIDNumber LIKE '90%' OR Title = 'Marketing Manager'
```

WHERE Clause Operators The operators you use in a `WHERE` clause affect how fast SQL Server runs a query. You might not have any choice of which operator you use in your `WHERE` clauses, but if you do, consider the following list, which is in the order of the fastest to the slowest:

```
=
>, >=, <, <=
LIKE
<>
```

Unless it is unavoidable, do not use `LIKE` or `<>`. Consider the following example:

```
USE AdventureWorks
GO
SELECT NationalIDNumber, Title
FROM HumanResources.Employee
WHERE ManagerID = 109
```

This `WHERE` clause will operate faster than the following one:

```
USE AdventureWorks
GO
SELECT NationalIDNumber, Title
FROM HumanResources.Employee
WHERE ManagerID LIKE '109'
```

Using LOWER and UPPER If you use the `LOWER` or `UPPER` functions in a `WHERE` clause, a table scan could take a long time because SQL Server carries out a case conversion on at least one column of each row during the scan—for example:

```
USE AdventureWorks
GO
SELECT LoginID
FROM HumanResources.Employees
WHERE LOWER(Title) = 'tool designer'
```

Typically, data in a SQL Server database is not case sensitive and you can simply omit the `LOWER` or `UPPER` functions. If the collation is case sensitive, you need to create an additional column in the table that is all uppercase or all lowercase, create an index for this column, and then specify the new column in the query. I described this in the Real World sidebar at the start of this chapter.

In general, you should avoid any conversions or calculations in WHERE clauses—for example, when finding the first letter of a word:

```
USE AdventureWorks
GO
SELECT Title, LoginID
FROM HumanResources.Employees
WHERE SUBSTRING(Title,1,1) = 'P'
```

Not only does this approach perform an unnecessary substring operation on every row, it also performs a table scan rather than use an index. The following query returns the same results more efficiently (even though the LIKE operator is slower than the = operator):

```
USE AdventureWorks
GO
SELECT Title, LoginID
FROM HumanResources.Employees
WHERE Title LIKE 'P%'
```

Monitoring CPU Utilization

You can use Profiler or stored procedures to generate a trace that records CPU usage for a query plan. If you do not want to run the query plan, you can use the Showplan event classes in Profiler to capture information about the estimated execution plan in XML or text format. You can also use the SSMS graphical execution plan to generate an actual or estimated graphical representation of the query's execution plan.

However, possibly the most comprehensive information about the CPU usage of an execution plan is provided by the *sys.dm_exec_query_stats* DMV. You need to run the query before the DMV returns the statistics and provides you with the following information:

- The number of times that the plan has been executed since it was last compiled
- The total amount of CPU time, in microseconds, that was consumed by executions of the plan since it was compiled
- The CPU time, in microseconds, that was consumed the last time the plan was executed
- The maximum CPU time, in microseconds, that the plan has ever consumed during a single execution
- The maximum CPU time, in microseconds, that the plan has ever consumed during a single execution

PRACTICE Obtaining Query Plan Statistics

In the following practice session, you obtain an estimated execution plan for a query against a table in the AdventureWorks database. The practice uses the SSMS graphical execution plan. You can extend the practice by using other methods described in this lesson to capture query plan statistics.

► Practice: Using the Management Studio Graphical Execution Plan to Analyze a Query Plan

1. Log in to your domain at your member server by using either your domain administrator account or an account that has been added to the sysadmin server role. If you are using virtual machine software, log in to your domain and connect to your member server.
2. From Programs, Microsoft SQL Server 2005, click SQL Server Management Studio. Connect to the database engine on your member server, specifying Windows Authentication. Specify TCP/IP and the AdventureWorks database. Click Connect.
3. Click New Query to start Query Editor.
4. In the Query pane, type the following query:

```
USE AdventureWorks
GO
SELECT * FROM HumanResources.Employee
WHERE SUBSTRING(NationalIDNumber,1,1) = '8'
UNION
SELECT *
FROM HumanResources.Employee
WHERE Title LIKE 'P%'GO
```

5. Click Display Estimated Execution Plan. The estimated execution plan output is shown in Figure 2-8. Right-click anywhere in the Execution Plan pane and choose Save Execution Plan As to save the estimated execution plan.

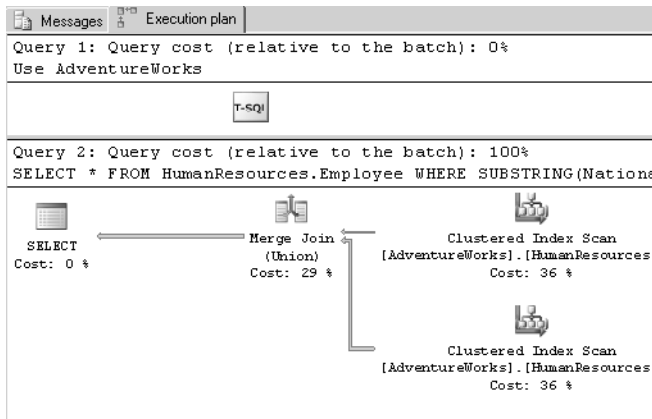


Figure 2-8 Estimated execution plan.

6. Place your mouse pointer over any element in the plan to obtain statistics for that element, as shown in Figure 2-9. The statistics you obtain might differ from those shown in the figure.

Clustered Index Scan
Scanning a clustered index, entirely or only a range.

Physical Operation	Clustered Index Scan
Logical Operation	Clustered Index Scan
Estimated I/O Cost	0.0075694
Estimated CPU Cost	0.000476
Estimated Operator Cost	0.0080454 (36%)
Estimated Subtree Cost	0.0080454
Estimated Number of Rows	181.25
Estimated Row Size	187 B
Ordered	True
Node ID	4

Predicate
[AdventureWorks],[HumanResources],[Employee],[Title]
like NP%

Object
[AdventureWorks],[HumanResources],[Employee].
[PK_Employee_EmployeeID]

Figure 2-9 Obtaining statistics for a plan element.

7. Use the same procedure to create an estimated execution plan for the following query:

```
USE AdventureWorks
GO
SELECT *
FROM HumanResources.Employee
WHERE NationalIDNumber LIKE '8%'
OR Title LIKE 'P%'
GO
```

8. The estimated execution plan is shown in Figure 2-10. Compare the two execution plans you generated. Both queries return the same data, but the second plan is simpler and more efficient.

Query 1: Query cost (relative to the batch): 0%
USE AdventureWorks

Query 2: Query cost (relative to the batch): 100%
SELECT * FROM HumanResources.Employee WHERE NationalIDNumber L...

Clustered Index Scan
[AdventureWorks].[HumanResources]. [...]
Cost: 100 %

Figure 2-10 The estimated execution plan for the more efficient query.

Lesson Summary

- You can display query execution plans by using Transact-SQL SET statements, SQL Server Profiler event classes, and the SSMS graphical execution plan.
- The *sys.dm_exec_query_stats* DMV returns statistics about a query execution plan, and it returns CPU utilization statistics.
- Excessive disk I/O operations and long query duration times can indicate a large number of table scans.
- You can reduce table scans by modifying your queries and by creating indexes.

Lesson Review

You can use the following questions to test your knowledge of the information in Lesson 2, “Analyzing a Query Plan to Detect Inefficiencies in Query Logic.” The questions are also available on the companion CD if you prefer to review them in electronic form.

NOTE Answers

Answers to these questions and explanations of why each answer choice is correct or incorrect are located in the “Answers” section at the end of the book.

- I. You are running a query against the Employees table in the HumanResources database. A clustered index exists on the Employee_Second_Name column of this table. The data in the table is case insensitive. You want to list all employees whose second name starts with M. You do not believe that any duplicate rows exist, and if they do, listing them does not cause a problem. Which of the following queries most quickly returns the information you need?
 - A.

```
SELECT Employee_Second_Name
FROM HumanResources.Employees
WHERE SUBSTRING(Employee_Second_Name,1,1) = 'M'
```
 - B.

```
SELECT Employee_Second_Name
FROM HumanResources.Employees
WHERE Employee_Second_Name LIKE 'm%'
```

- C. `SELECT DISTINCT Employee_Second_Name`
`FROM HumanResources.Employees`
`WHERE SUBSTRING(Employee_Second_Name,1,1) = 'M'`
- D. `SELECT DISTINCT Employee_Second_Name`
`FROM HumanResources.Employees`
`WHERE Employee_Second_Name LIKE 'm%'`
2. You are analyzing a query plan for a query that you believe is causing excessive logical read operations. You do not want to use resources on the production server, and you have no test or monitoring server. You do not have a problem with executing the query to analyze its plan. You need to find out the following information as quickly as possible:
- ❑ The total number of logical reads performed by executions of this plan since it was compiled
 - ❑ The number of logical reads performed the last time the plan was executed
 - ❑ The minimum and maximum number of logical reads that this plan has ever performed during a single execution

How should you proceed?

- A. Capture a trace by using Profiler. Run the DTA specifying this trace as a workload.
- B. Copy the query into Query Editor. Access the DTA directly from SSMS.
- C. Copy the query into Query Editor. Use the SSMS graphical execution plan facility to graphically display the data retrieval methods chosen by the SQL Server query optimizer.
- D. Use the `sys.dm_exec_query_stats` DMV.

Lesson 3: Maintaining and Optimizing Indexes

A database index is a pointer to data in a table. It directs a query to the exact physical location of data in that table. Technically, you are being directed to the data's location in an underlying database, but in effect you are referring to a table.

After this lesson, you will be able to:

- Defragment an index.
- Rebuild an index.
- Specify a fill factor.
- Know when to enable the pad index option.
- Use clustered and nonclustered indexes.
- Use covering indexes.
- Use indexed views.
- Use included columns and composite indexes.
- Use XML indexes.
- Use partitioned indexes.

Estimated lesson time: 60 minutes

Defragmenting an Index

The SQL Server 2005 database engine automatically maintains indexes whenever INSERT, UPDATE, or DELETE operations occur. Over time, these modifications can cause the information in the index to become fragmented. Fragmentation occurs when indexes have pages in which the logical order—based on the key value—does not match the physical order inside the data file. Heavily fragmented indexes can degrade query performance and cause applications to respond slowly. You can remedy index fragmentation by either reorganizing or rebuilding an index.

The first step in deciding which method to use to defragment an index is to determine the degree of fragmentation. You can use the system function `sys.dm_db_index_physical_stats` to detect fragmentation in a specific index, in all indexes on a table or in an indexed view, in all indexes in a database, or in all indexes in all databases. The percentage of logical fragmentation is displayed in the `avg_fragmentation_in_percent` column. For example, if you were writing a stored procedure to check the indexes in a database for fragmentation, you would use the following Transact-SQL statement:

```
SELECT * FROM sys.dm_db_index_physical_stats
```

The preceding expression is a statement, not a query or procedure, and you need to specify arguments such as database identity, object identity, index identity, partition number, and node. The following is a typical query that uses the DMV:

```
USE master;
GO
SELECT * FROM sys.dm_db_index_physical_stats
    (DB_ID(N'AdventureWorks'), OBJECT_ID(N'Person.Address'),
    NULL, NULL, 'DETAILED');
GO
```

MORE INFO *sys.dm_db_index_physical_stats* arguments

For more information about the use of *sys.dm_db_index_physical_stats* and how to specify its arguments, search Books Online for “*sys.dm_db_index_physical_stats*” or access [msdn2.microsoft.com/en-us/library/ms188917\(d=ide\).aspx](https://msdn2.microsoft.com/en-us/library/ms188917(d=ide).aspx).

NOTE Use the *sys.dm_db_index_physical_stats* DMV.

Microsoft recommends using the *sys.dm_db_index_physical_stats* DMV rather than DBCC SHOWCONTIG to detect fragmentation.

For partitioned indexes, *sys.dm_db_index_physical_stats* also provides fragmentation information for each partition. The result set returned by *sys.dm_db_index_physical_stats* includes the columns listed in Table 2-1.

Table 2-1 *sys.dm_db_index_physical_stats* Columns

Column	Description
avg_fragmentation_in_percent	The percent of logical fragmentation (out-of-order pages in the index)
fragment_count	The number of fragments (physically consecutive leaf pages) in the index
avg_fragment_size_in_pages	The average number of pages in a single fragment in an index

Reorganizing and Rebuilding an Index

After you have determined the degree of fragmentation, you can use ALTER INDEX REORGANIZE to correct the fragmentation if it is less than or equal to 30 percent. Otherwise, you should use ALTER INDEX REBUILD WITH (ONLINE = ON).

IMPORTANT Reorganizing and rebuilding

Reorganizing an index is always executed online. Rebuilding an index can be executed online or offline.

Reorganizing an Index

To reorganize one or more indexes, use the ALTER INDEX statement with the REORGANIZE clause. This statement replaces the DBCC INDEXDEFRAG statement. To reorganize a single partition of a partitioned index, use the PARTITION clause of ALTER INDEX.

Exam Tip Examiners frequently test candidates' knowledge of new features. Sometimes the method used in a previous version of the software is given as a distracter (wrong answer). The purpose is to test whether candidates with (say) SQL Server 2000 experience have learned the new procedures that SQL Server 2005 provides.

Reorganizing an index defragments the leaf level (the lowest level) of clustered and nonclustered indexes on tables and views by physically reordering the leaf-level pages to match the logical order (left to right) of the leaf nodes. Putting the pages in order improves index-scanning performance. The index is reorganized within the existing pages allocated to it, and no new pages are allocated. If an index spans more than one file, the files are reorganized one at a time. Pages do not migrate between files.

Reorganizing also compacts the index pages. Any empty pages created by this compaction are removed, providing additional available disk space. Compaction is based on the fill factor value in the sys.indexes catalog view. (This lesson discusses the fill factor in more detail later.) The reorganize process uses minimal system resources. Also, reorganizing is automatically performed online and does not hold long-term blocking locks. The process does not block running queries or updates. You should reorganize an index when it is not heavily fragmented. Otherwise, you achieve better results by rebuilding the index.

Rebuilding an Index

Rebuilding an index drops the index and creates a new one. This process removes fragmentation, reclaims disk space by using the specified or existing fill factor setting to compact the pages, and reorders the index rows in contiguous pages (allocating new pages as needed). Rebuilding an index improves disk performance by reducing the

number of page reads required to obtain requested data. The following methods can be used to rebuild clustered and nonclustered indexes:

- ALTER INDEX with the REBUILD clause (replaces DBCC DBREINDEX)
- CREATE INDEX with the DROP_EXISTING clause

Both methods do the same job but have differing functionality, as shown in Table 2-2.

Table 2-2 Differences Between Rebuild Methods

Functionality	ALTER INDEX REBUILD	CREATE INDEX WITH DROP_EXISTING
Index definition can be changed by adding or removing key columns, changing column order, or changing the column sort order.	No	Yes
Index options can be set or modified.	Yes	Yes
More than one index can be rebuilt in a single transaction.	Yes	No
Most index types can be rebuilt online without blocking running queries or updates.	Yes	Yes
Partitioned index can be repartitioned.	No	Yes
Index can be moved to another file-group.	No	Yes
Additional temporary disk space is required.	Yes	Yes
Rebuilding a clustered index rebuilds associated nonclustered indexes.	No, unless the ALL keyword is specified	No, unless the index definition is changed
Indexes enforcing PRIMARY KEY and UNIQUE constraints can be rebuilt without dropping and re-creating the constraints.	Yes	Yes
Single index partition can be rebuilt.	Yes	No

NOTE Using separate DROP and CREATE statements

You can also rebuild an index by first dropping it with the DROP INDEX statement and re-creating it with a separate CREATE INDEX statement. However, Microsoft does not recommend this procedure.

Adding an Index

You can create an index by using the CREATE INDEX Transact-SQL statement. This creates a relational index on a specified table or view, or an XML index on a specified table. You can create an index before there is data in the table. You can create indexes on tables or views in another database by specifying a qualified database name. You can create unique, clustered, and nonclustered indexes and specify whether the index uses ascending or descending sort direction for the specified index column.

MORE INFO Creating an index

For more information, search for "CREATE INDEX" in Books Online or access msdn.microsoft.com/library/default.asp?url=/library/en-us/tsqlref/ts_create_6414.asp.

Creating or modifying a table and specifying the PRIMARY KEY or UNIQUE constraint enforces the uniqueness of rows and implicitly creates an index to do so. This index is clustered by default. A table typically has a column or combination of columns that contains values that uniquely identify each row in the table. This column, or columns, is called the primary key (PK) of the table and enforces the entity integrity of the table.

You can create a PK by defining a PRIMARY KEY constraint when you create or modify a table. A table can have only one PRIMARY KEY constraint, and a column that participates in the PRIMARY KEY constraint cannot accept null values. Because PRIMARY KEY constraints guarantee unique data, they are frequently defined on an identity column.

When you specify a PRIMARY KEY constraint for a table, the SQL Server 2005 database engine enforces data uniqueness by creating a unique index for the primary key columns. This index also permits fast access to data when the PK is used in queries. Therefore, the PKs that are chosen must follow the rules for creating unique indexes. If you define a PRIMARY KEY constraint on more than one column, values can be duplicated within one column, but each combination of values from all the columns in the PRIMARY KEY constraint definition must be unique.

When you specify a UNIQUE constraint, a unique nonclustered index is created to enforce the constraint. You can specify a unique clustered index if a clustered index on the table does not already exist. You can also create a unique index independent of a constraint and define multiple unique nonclustered indexes on a table. To create an indexed view, you define a unique clustered index on one or more view columns. The view is executed (materialized), and the result set is stored in the leaf level of the index in the same way table data is stored in a clustered index.

Specifying the Fill Factor

When you create or rebuild an index, the fill factor value determines the percentage of space on each leaf-level page that SQL Server fills with data, and it reserves a percentage of free space for future growth. You specify the fill factor value as a percentage from 1 to 100. In most situations, you would not change the default of 0 (which actually means 100 percent). With this fill factor, SQL Server fills the leaf-level pages almost to capacity, but some space remains for at least one additional index row on each page.

You can use the CREATE INDEX or ALTER INDEX statements to set the fill factor value for individual indexes, and the sys.indexes catalog view to view the fill factor value of one or more indexes. The *sp_configure* system stored procedure can modify the server-wide default fill factor value. Fill factor is an advanced option. If you use this stored procedure, you can change the fill factor setting only when you set Show Advanced Options to 1. The setting takes effect after you restart the server.

IMPORTANT The fill factor is not maintained

The fill factor setting applies only when an index is created or rebuilt. The SQL Server 2005 database engine does not dynamically keep the specified percentage of empty space in the pages.

Page Splits

When SQL Server adds a new row to a full index page, the database engine moves approximately half the rows to a new page to make room. This reorganization is known as a page split. A page split can take time to perform, is a resource-intensive operation, can cause fragmentation, and results in an increase in disk I/O operations. A correctly chosen fill factor value reduces the likelihood of page splits by providing enough space for index expansion as data is added to the underlying table.

CAUTION Take care when specifying a low fill factor

A fill factor other than 0 reduces the requirement to split pages as the index grows, but it requires more storage space and can decrease read performance. Typically, database reads outnumber database writes by a factor of 5 through 10. Therefore, specifying a low fill factor that increases the number of reads can dramatically affect database performance. If, however, a table is write-intensive, a high number of queries run against it, and it becomes fragmented shortly after it is rebuilt, you should consider rebuilding the table indexes with a lower fill factor—for example 75 percent.

Using the PAD_INDEX Option

The *pad index* specifies the space left free on each page in the intermediate levels of an index. Pad index is disabled unless you specify a fill factor. If you set the PAD_INDEX option of CREATE INDEX to ON, SQL Server applies the percentage of free space you specify by the FILLFACTOR option to the intermediate-level pages of the index. If you set the PAD_INDEX option to OFF, SQL Server fills the intermediate-level pages of the index to near capacity, leaving sufficient space for at least one row of the maximum size the index can have.

If the specified fill factor does not allow sufficient space for one row, the database engine overrides the fill factor percentage to allow this space. The number of rows on an intermediate index page is never fewer than two, regardless of the fill factor value.

You should consider using the FILLFACTOR and PAD_INDEX options on tables that experience high levels of updates on indexed columns. Otherwise, extensive index updates can cause the intermediate index pages to split frequently, resulting in unnecessary overhead. If, however, your tables do not experience a lot of update activity, do not enable pad index. Leaving unnecessary free space on your intermediate pages results in an increased number of pages and increased disk I/O when SQL Server reads these pages. As with the fill factor, you specify the pad index setting when you create or rebuild the index.

Using Clustered and Nonclustered Indexes

An index contains keys built from one or more columns in the table or view. These keys are stored in a tree structure known as a *b-tree* that enables SQL Server to find the row or rows associated with the key values quickly and efficiently. A table or view can contain clustered or nonclustered indexes.

Clustered Indexes

Clustered indexes sort and store the data rows in the table or view based on their key values. These keys are the columns you included in the index definition. Only one clustered index can exist on a table because SQL Server can sort the data rows in only one order. You therefore need to choose the clustered index column carefully. By default, SQL Server automatically creates a clustered index based on the primary key, but you might want to specify another column depending on the queries you expect to run against the table. SQL Server can store data rows in a table in order only when the table contains a clustered index. A table that has a clustered index is called a *clustered table*. If a table has no clustered index, its data rows are stored in an unordered structure called a *heap*.

When you create an index with the CREATE INDEX statement, you can use the CLUSTERED option to create a clustered index in which the logical order of the key values determines the physical order of the corresponding rows in a table. The bottom (or leaf) level of the clustered index contains the actual data rows of the table. You can also create a view and specify a clustered index. A view with a unique clustered index is called an *indexed view*. You must create a unique clustered index on a view before you can define any other indexes on the same view. The section “Using Indexed Views” later in this chapter discusses indexed views in more detail.

MORE INFO Designing indexed views

For more information, search for “Designing Indexed Views” in Books Online or access [msdn2.microsoft.com/en-us/library/ms187864\(d=ide\).aspx](https://msdn2.microsoft.com/en-us/library/ms187864(d=ide).aspx).

You should always create the clustered index before creating any nonclustered indexes. This is because SQL Server automatically rebuilds any existing nonclustered indexes on tables when you create a clustered index. If you do not specify CLUSTERED when you create an index, SQL Server creates a nonclustered index.

In SQL Server, indexes are organized as b-trees. Each page in an index b-tree is called an *index node*, and the top node is called the *root node*. The nodes at the lowest level in the index are called *leaf nodes*. Any index levels between the root and leaf nodes are known as *intermediate levels*. In a clustered index, the leaf nodes contain the data pages of the underlying table. The root and leaf nodes contain index pages holding index rows. Each index row contains a key value and a pointer to either an intermediate level page in the b-tree or a data row in the leaf level of the index.

Clustered indexes have one row in the Transact-SQL object catalog view `sys.partitions`, with `index_id = 1` for each partition used by the index. By default, a clustered index has a single partition. If it has multiple partitions, each partition has a b-tree structure that contains the data for that specific partition.

Depending on the data types in the clustered index, each clustered index has one or more allocation units in which to store and manage the data for a specific partition. At a minimum, each clustered index has one `IN_ROW_DATA` allocation unit per partition. The clustered index also has one `LOB_DATA` allocation unit per partition if it contains LOB columns.

MORE INFO Allocation units

For more information about allocation units, search for "Table and Index Organization" in Books Online or access [msdn2.microsoft.com/en-us/library/ms189051\(d=ide\).aspx](https://msdn2.microsoft.com/en-us/library/ms189051(d=ide).aspx).

Nonclustered Indexes

Nonclustered indexes have a structure separate from the data rows. A nonclustered index contains nonclustered index key values, and each key value entry has a pointer to the data row in the actual table that contains the key value. The pointer from an index row in a nonclustered index to a data row is called a *row locator*. The structure of the row locator depends on whether the data pages are stored in a heap or clustered table. For a heap, a row locator is a pointer to the row and is known as a *row identity* (RID). For a clustered table, the row locator is the clustered index key.

If you use the `NONCLUSTERED` option in the `CREATE INDEX` statement or create a nonclustered index implicitly with the `PRIMARY KEY` and `UNIQUE` constraints, the index specifies the logical ordering of a table. If an index is nonclustered, the physical order of the data rows is independent of their indexed order. Each table can have up to 249 nonclustered indexes, regardless of how the indexes are created.

Nonclustered indexes have the same b-tree structure as clustered indexes, except that the data rows of the underlying table are not sorted and stored in an order based on their nonclustered keys, and the leaf layer of a nonclustered index is made up of index pages instead of data pages. You can define nonclustered indexes on a table or view with a clustered index, or on a heap. Each index row in the nonclustered index contains the nonclustered key value and a row locator. This locator points to the data row in the clustered index or heap having the key value.

Nonclustered indexes have one row in `sys.partitions` with `index_id > 1` for each partition used by the index. By default, a nonclustered index has a single partition. When a nonclustered index has multiple partitions, each partition has a b-tree structure that contains the index rows for that specific partition.

Composite Indexes and Included Columns

You can specify a column for a clustered index and then specify *included columns* to create a composite index. If you specify two or more column names, you create a composite index on the combined values in the specified columns. In the `CREATE INDEX` Transact-SQL statement, you should list the columns to be included in the composite index, in sort-priority order, inside the parentheses after `table_or_view_name`. You can combine up to 16 columns into a single composite index key. All the columns in a composite index key must be in the same table or view. The maximum allowable size of the combined index values is 900 bytes.

Normalization

The logical design and optimization of a database—including the design of tables, indexes, views, constraints, and table joins—is part of a process known as *normalization*. A good logical database design can lay the foundation for optimal database and application performance. Normalizing a logical database design involves using formal methods to separate the data into multiple, related tables. Several narrow tables with fewer columns are characteristic of a normalized database. A few wide tables with more columns are characteristic of a non-normalized database. Reasonable normalization frequently improves performance. When useful indexes are available, the SQL Server query optimizer is efficient at selecting rapid, efficient joins between tables.

Using Covering Indexes

If you use a nonclustered index and you know that your application will be performing the same query many times on the same table, you should consider creating a covering index on the table. A covering index includes all the columns referenced in the `SELECT`, `JOIN`, and `WHERE` clauses of a query. As a result, the index contains the data for which you are looking and your query does not need to access the table to obtain the data you require. The rows of the covering index contain a subset of the

columns in the table; thus, there are more rows per page. As a result, SQL Server needs to perform fewer paging operations to retrieve the query data, boosting performance.

If, on the other hand, the covering index covers too many columns, disk I/O might be increased and performance degraded. You should use covering indexes only if the query or queries are run frequently. Otherwise, the overhead of maintaining the covering index might outweigh the benefits it provides. The covering index should not add significantly to the size of the key and must include all columns found in the SELECT list, JOIN clause, and WHERE clause.

If you want to create a covering index, try if possible to incorporate already existing table indexes. For example, if you need a covering index for the Price and Description columns, and you already have an index on the Price column, you should change the index on the Price column to a composite index on Price and Description instead of creating a new covering index. In general, try not to create more than one index on the same column.

An alternative to creating covering indexes is to let SQL Server 2005 create the covering indexes for you. The query optimizer can perform *index intersection*. The optimizer considers multiple indexes from a table, builds a hash table based on the multiple indexes, and then uses the hash table to reduce the I/O the query requires. In effect, the hash table becomes a covering index for the query. If you create single column, nonclustered indexes on all the columns in a table that will be queried frequently, the query optimizer will be provided with the data it needs to create covering indexes as required.

Quick Check

- A SQL Server 2005 database contains a table named Products.Electrical that has more than 30 columns. Typically, queries against that table are of the following form:

```
SELECT Name, Price, Rating, Class, Color
FROM Products.Electrical
WHERE Price < 10000
ORDER BY Name
```

Which two indexes should you create?

Quick Check Answer

- Create a clustered index on the Name column. Create a nonclustered index on the Price column, and include the Rating, Class, and Color columns. You might also consider creating a covering index in this scenario.

Using Indexed Views

A view can contain columns from one or more tables. SQL Server does not store the result set of a standard or nonindexed view permanently in the database. Every time a query references a standard view, SQL Server 2005 substitutes the definition of the view into the query until a modified query is formed that references only the standard view's base tables. The modified query then runs in the normal way. Views are also known as virtual tables because the result set returned by the view has the same general form as a table with columns and rows, and you can reference views in the same way as tables in SQL statements.

The overhead of dynamically building the result set for each query that references a standard view can be significant. If you frequently reference such views in queries, you can improve performance by creating a unique clustered index on the view. When you create a unique clustered index on a view, SQL Server stores the result set as a separate object within the database. This improves performance because SQL Server does not have to retrieve the view's result set each time a query references this view. Another benefit of creating an index on a view is that the query optimizer starts using the view index in queries that do not directly name the view in the FROM clause. As users make modifications to the data in the base tables, SQL Server also updates the data stored in the indexed view. The clustered index of the view must be unique, which improves the efficiency with which SQL Server can find the rows in the index that are affected by any data modification. The disadvantage of using indexed views is that updating an indexed view can increase the workload on your SQL Server 2005 server. Typically, you create indexed views on tables that contain information that is read-intensive and is updated infrequently.

The SQL Server 2005 query processor treats indexed and standard views differently. The rows of an indexed view are stored in the database in the same format as a table. If the query optimizer decides to use an indexed view in a query plan, the indexed view is treated the same way as a base table. For a standard view, only the definition is stored, not the rows of the view. The query optimizer incorporates the logic from the view definition into the execution plan it builds for the SQL statement that references the standard view.

MORE INFO Indexed views

For more information, search for "Creating Indexed Views," "View Resolution," and "Resolving Indexes on Views" in Books Online or access [msdn2.microsoft.com/en-us/library/ms191432\(d=ide\).aspx](http://msdn2.microsoft.com/en-us/library/ms191432(d=ide).aspx), [msdn2.microsoft.com/en-us/library/ms190237\(d=ide\).aspx](http://msdn2.microsoft.com/en-us/library/ms190237(d=ide).aspx), and [msdn2.microsoft.com/en-us/library/ms181151\(d=ide\).aspx](http://msdn2.microsoft.com/en-us/library/ms181151(d=ide).aspx). Also, read the white paper "Improving Performance with SQL Server 2005 Indexed Views," which is available at www.microsoft.com/technet/prodtechnol/sql/2005/ipsql05iv.mspx.

Creating XML Indexes

SQL Server stores instances of XML data as binary large objects (BLOBs), which can be up to 2 GB in size. If you do not create an index on the XML columns, evaluating a BLOB at run time can be time-consuming. However, a cost is associated with maintaining the index during data modification. XML indexes fall into two categories:

- Primary XML index
- Secondary XML index

The primary XML index is the first index on the xml type column. Three types of secondary indexes are supported: PATH, VALUE, and PROPERTY. The primary index creates several rows of data for each XML BLOB in the column. The number of rows in the index is approximately equal to the number of nodes in the XML BLOB. Each row stores the following information:

- Tag name such as an element or attribute name.
- Node value.
- Node type, such as an element node, attribute node, or text node.
- Document order information, represented by an internal node identifier.
- Path from each node to the root of the XML tree. This column is searched for path expressions in the query.
- Primary key of the base table. The primary key of the base table is duplicated in the primary XML index for back join with the base table, and the maximum number of columns in the primary key of the base table is limited to 15.

SQL Server uses this node information to evaluate and construct XML results for a specified query. Paths are stored in reverse order to allow paths to be matched when only the path suffix is known. The query processor uses the primary XML index for queries involving XML data type methods and returns either scalar values or the XML subtrees from the primary index itself.

MORE INFO XML data type methods

For more information, search for “xml data type methods” in Books Online or access [msdn2.microsoft.com/en-us/library/ms190798\(d=ide\).aspx](https://msdn2.microsoft.com/en-us/library/ms190798(d=ide).aspx).

You can create secondary XML indexes to enhance search performance. Before you can create secondary indexes, a primary XML index must first exist. The following list shows the types of secondary XML indexes:

- PATH

- VALUE
- PROPERTY

A primary XML index might not provide the best performance for queries based on path expressions, and if your queries typically specify path expressions on xml type columns, a PATH secondary index might speed up the search. Because all rows in the primary XML index corresponding to an XML BLOB are searched sequentially for large XML instances, the search might be slow. If that is the case, a secondary index built on the path values and node values in the primary index can significantly speed up the index search.

If your queries are value based and the path is not fully specified or includes a wildcard, you might obtain faster results by building a secondary XML index that is built on node values in the primary XML index. The key columns of the VALUE index are the node value and path of the primary XML index. If your workload involves querying for values from XML instances without knowing the element or attribute names that contain the values, the VALUE index might be useful.

Queries that retrieve one or more values from individual XML instances might benefit from a PROPERTY index. This scenario occurs when you retrieve object properties using the value() method of the xml type and when the primary key value of the object is known. The PROPERTY index is built on columns (PK, Path, and node value) of the primary XML index, where PK is the primary key of the base table.

Creating an XML index on an xml type column is similar to creating an index on a non-xml type column. You can use the following Transact-SQL statements to create and manage XML indexes:

- CREATE INDEX
- ALTER INDEX
- DROP INDEX

Creating a Primary XML Index

The Transact-SQL statement CREATE PRIMARY XML INDEX creates a primary XML index. The base table that contains the XML column you are indexing must have a clustered index on the primary key. This ensures that if the base table is partitioned, the primary XML index can be partitioned by using the same partitioning scheme and partitioning function.

You can create a primary XML index on a single xml type column, and each xml type column in a table can have its own primary XML index. You cannot create any other

type of index with the XML type column as a key column. However, you can include the xml L type column in a non-XML index. Only one primary XML index per xml type column is permitted.

Creating a Secondary XML Index

You can use the Transact-SQL CREATE XML INDEX statement to create secondary XML indexes and specify their type. All indexing options that apply to a nonclustered index, except IGNORE_DUP_KEY and ONLINE, are permitted on secondary XML indexes. These two options must always be set to OFF for secondary XML indexes.

You can query the sys.xml_indexes catalog view to retrieve XML index information. The secondary_type_desc column in the sys.xml_indexes catalog view provides the type of secondary index. The values returned in the secondary_type_desc column can be NULL, PATH, VALUE, or PROPERTY. For a primary XML index, the value returned is NULL.

Creating Partitioned Indexes

Partitioning makes large tables or indexes more manageable because it lets you manage and access subsets of data quickly and efficiently while maintaining the integrity of a data collection. Maintenance operations on subsets of data are also performed more efficiently because they target only the data that is required.

IMPORTANT Supported in Enterprise and Developer editions only

Partitioned tables and indexes are supported in the SQL Server 2005 Enterprise and Developer editions only.

SQL Server divides the data of partitioned tables and indexes into units that you can spread across more than one filegroup in a database. SQL Server partitions the data horizontally so that groups of rows are mapped into individual partitions. The table or index is treated as a single logical entity when you perform queries or updates on the data. All partitions of a single index or table must reside in the same database.

To create a partitioned index, you must first create a partition function and a partition scheme. The partition function, created using the Transact-SQL CREATE PARTITION FUNCTION statement, specifies how an index (or a table) that uses the function can be partitioned. The partition scheme, created using the Transact-SQL CREATE PAR-

TITION SCHEME statement, specifies the placement of the partitions of a partition function on filegroups.

MORE INFO Creating partition functions and schemes

For more information, search Books Online for "CREATE PARTITION FUNCTION" and "CREATE PARTITION SCHEME" or access [msdn2.microsoft.com/en-us/library/ms187802\(SQL.90,d=ide\).aspx](https://msdn2.microsoft.com/en-us/library/ms187802(SQL.90,d=ide).aspx) and [msdn2.microsoft.com/en-us/library/ms179854\(SQL.90,d=ide\).aspx](https://msdn2.microsoft.com/en-us/library/ms179854(SQL.90,d=ide).aspx).

You use the ON clause of the Transact-SQL CREATE INDEX statement to specify the partition scheme name and the column against which the index will be partitioned. This column must match the data type, length, and precision of the argument of the partition function. You can specify any column in the base table, except that when you partition a UNIQUE index, you must choose the column from among those used as the unique key.

If you do not use the ON clause and the table on which you are creating an index is partitioned, SQL Server places the index in the same partition scheme and uses the same partitioning column as the underlying table. Although you can implement partitioned indexes independently from their base tables, it generally makes sense to design a partitioned table and then create an index on the table. This approach results in the index being partitioned in the same manner as the table and aligns the index with the table.

NOTE Tuning aligned indexes by using the DTA

The Tuning Options tab of the DTA provides an Aligned Partitioning setting to specify that new recommended indexes be aligned with their base tables. The Keep Aligned Partitioning setting can be used for the same purpose and can also be used to drop existing nonaligned indexes.

PRACTICE Performing Index Analysis

In the following practice session, you tune the indexes on a table in the AdventureWorks database. You can extend the practice by re-creating an existing index or creating a new nonclustered index on the table and performing the analysis again. In this practice, you analyze the workload generated by a single query. Analyzing a complex workload generated by a period of production activity would generate more recommendations and reports.

► **Practice: Using the DTA to Tune Aligned Indexes**

To use the DTA to tune indexes, complete the following steps:

1. Log in to your domain at your member server by using either your domain administrator account or an account that has been added to the sysadmin server role. If you are using virtual machine software, log in to your domain and connect to your member server.
2. From the Programs (or All Programs) menu, choose Microsoft SQL Server 2005, Performance Tools, and then Database Engine Tuning Advisor. Connect to your member server and the AdventureWorks database. On the File menu, choose New Session.
3. On the General tab, specify a workload file such as the SlowQueryTrace you created earlier, ensure the AdventureWorks database is selected, and then select the Employee table, as shown in Figure 2-11.

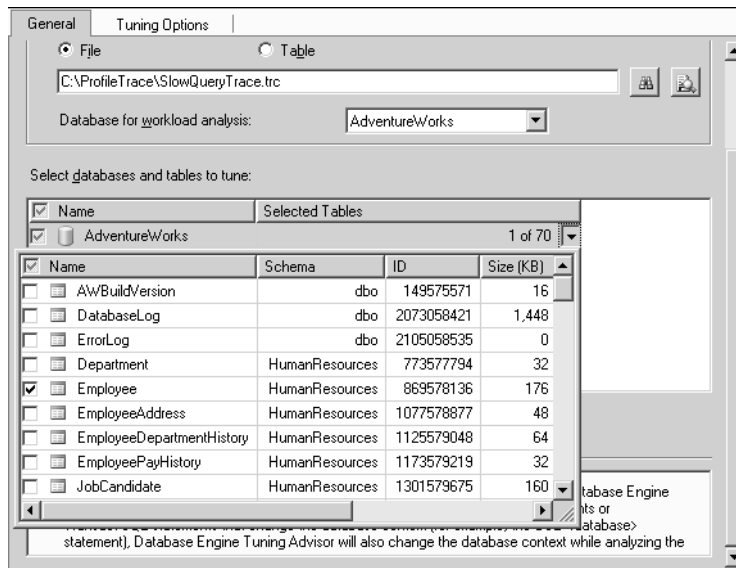


Figure 2-11 Selecting the workload and database table.

4. On the Tuning Options tab, limit the tuning time as shown in Figure 2-12.
5. On the toolbar, click Start Analysis.

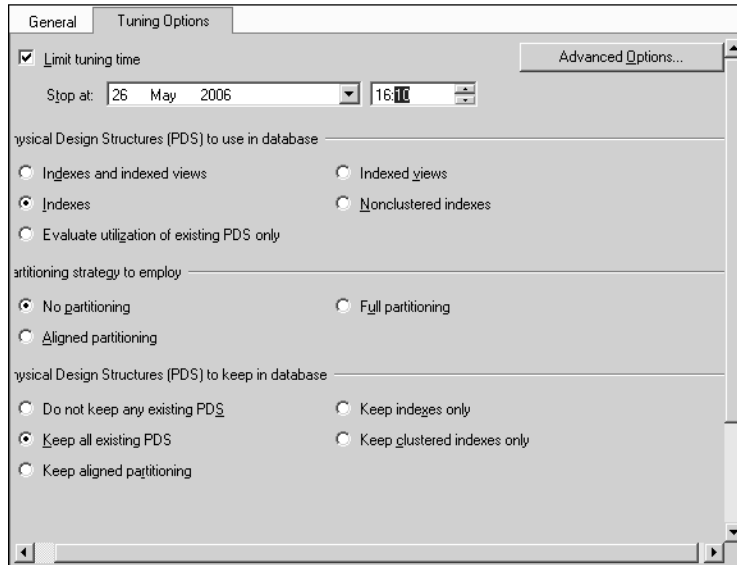


Figure 2-12 Selecting the tuning options.

6. Access the DTA output when the analysis completes. Figure 2-13 shows the Progress tab. The results you obtain vary depending upon the workload file you specify, and the DTA does not make recommendations unless significant improvements can be made.

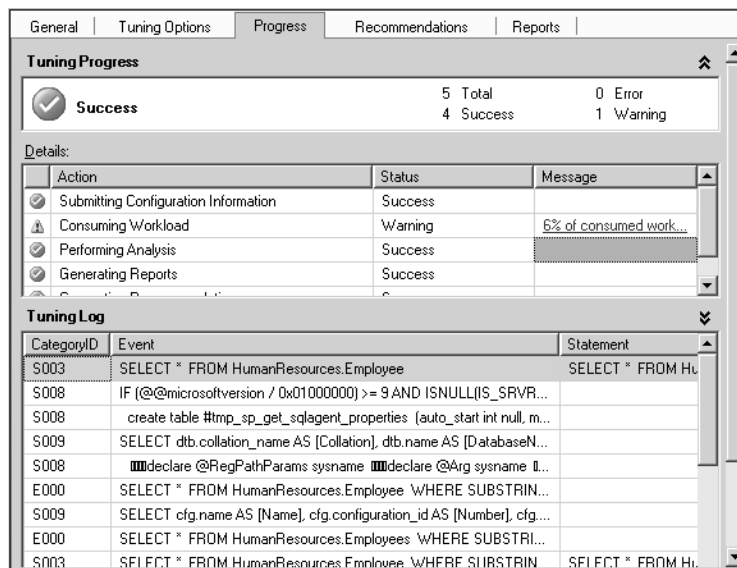


Figure 2-13 DTA analysis output.

Lesson Summary

- You can remedy index fragmentation by either reorganizing or rebuilding an index.
- You can address page split problems by specifying a fill factor and enabling the pad index option.
- You can improve query performance by creating a clustered index and one or more unclustered indexes on a table, and by creating an indexed view.
- If you use a nonclustered index and you know that your application will be performing the same query many times on the same table, you should consider creating a covering index on the table.
- You can improve the performance of queries that run against tables that contain BLOBs by creating XML indexes.
- Partitioning makes large tables or indexes more manageable because it lets you manage and access subsets of data quickly and efficiently while maintaining the integrity of a data collection.

Lesson Review

You can use the following questions to test your knowledge of the information in Lesson 3, “Maintaining and Optimizing Indexes.” The questions are also available on the companion CD if you prefer to review them in electronic form.

NOTE Answers

Answers to these questions and explanations of why each answer choice is correct or incorrect are located in the “Answers” section at the end of the book.

1. You need to find out the degree of fragmentation of a table index. How do you do this?
 - A. Use DBCC INDEXDEFRAG.
 - B. Use the *sys.dm_db_index_physical_stats* DMV.
 - C. Use the *sys.indexes* catalog view.
 - D. Use the CREATE INDEX Transact-SQL statement with the DROP_EXISTING clause.

2. Your company uses a SQL Server 2005 database that contains a table named Sales.Item. The table has 12 columns. The most common queries that are run against the table take the following form:

```
SELECT Name, ProductLine, Price, Color, Style
FROM Sales.Item
WHERE ProductLine = '<product_line_name>'
ORDER BY Name
```

Which indexes should you create? (Choose all that apply).

- A. Create a nonclustered index on the Name column.
 - B. Create a clustered index on the Name column.
 - C. Create a nonclustered index on the ProductLine column, and include the Price, Color, and Style columns.
 - D. Create a clustered index on the ProductLine, Price, Color, and Style columns.
3. A query runs frequently against a database. It specifies the same 3 columns of a 40-column table and the same 4 columns of a 25-column table. Both tables are read-intensive and updated infrequently. How do you optimize the query?
- A. Create a standard view that contains all the columns that are used in the query.
 - B. Rebuild the indexes on both tables, and enable the pad index option.
 - C. Create an indexed view that contains all the columns that are used in the query.
 - D. Rebuild the indexes on the larger table, and specify a fill factor of 75%.

Lesson 4: Enforcing Appropriate Stored Procedure Logging and Output

When you trace a stored procedure, information is written in a trace log. You need to ensure that sufficient useful information is available to help you troubleshoot any problems, to ensure that logs do not contain too much information, and to ensure that the logs are not difficult to interpret.

After this lesson, you will be able to:

- Ensure logging does not produce too little output.
- Ensure logging does not produce too much output.
- Configure exception handling.

Estimated lesson time: 30 minutes

Using Stored Procedures

A stored procedure is an executable object stored in a database that consists of one or more SQL statements compiled into a single executable procedure. SQL Server 2005 also supports extended stored procedures, which are C or C++ dynamic-link libraries (DLLs) written to the SQL Server Open Data Services application program interface (API) for extended stored procedures. Calling a stored procedure on the data source (instead of executing or preparing a statement in the client application directly) can provide higher performance, reduced network overhead, better consistency and accuracy, and additional functionality.

You can use the Default Trace Enabled option in Profiler to enable or disable the trace log files (enabled by default). The default trace functionality provides a rich, persistent log of activity and changes that are primarily related to the configuration options. Default trace provides troubleshooting assistance to DBAs by ensuring that they have the log data necessary to diagnose problems the first time they occur.

You can open and examine the default trace logs by using Profiler, or you can query them by using the Transact-SQL *fn_trace_gettable* system function. The default trace log is stored by default in the \MSSQL\LOG directory using a rollover trace file. The file name for the default trace log file is log.trc. The following statement opens the default trace log in the default location:

```
SELECT *
FROM fn_trace_gettable
('C:\Program Files\Microsoft SQL Server\MSSQL.1\MSSQL\LOG\log.trc', default)
GO
```


The Default Trace Enabled option enables a default trace when it is set to 1 (the default). You can use the *sp_configure* system stored procedure to change the setting only when Show Advanced Options is enabled. Trace logs contain a lot of information about report server operations, including redundant information recorded in other log files and additional information that is not otherwise available. Trace logs are useful if you are investigating a specific problem that was recorded in the event log or the execution log.

Viewing Log Information

You can trace logs in Profiler. Trace logs contain the following information:

- System information, including operating system, version, number of processors, and memory
- Version information
- Application log events
- Exceptions
- Low resource warnings
- HTTP header, stack trace, and debug trace information

Managing Log File Content

By default, trace logs are limited to 32 megabytes in size and deleted after 14 days. The events recorded in a trace file are instances of the event classes in the trace definition. Event classes and their event categories are available on the Events Selection tab of the Trace File Properties dialog box in Profiler. To obtain relevant data from a trace file, you can examine the Errors And Warnings event category. This category includes event classes that are produced when a SQL Server error or warning is returned. The Performance event category is also useful. This category includes event classes that are produced when DML operators (for example, DELETE, INSERT, and UPDATE) execute. The Showplan Statistics event class in the Performance event category can provide you with useful information about query and stored procedure operations.

MORE INFO Event classes

For more information, search for "SQL Server Event Class Reference" in Books Online or access [msdn2.microsoft.com/en-us/library/ms175481\(d=ide\).aspx](https://msdn2.microsoft.com/en-us/library/ms175481(d=ide).aspx). Follow the links to "Errors and Warnings Event Category (Database Engine)," "Performance Event Category," and "Performance Statistics Event Class."

Handling Exceptions

Run-time exceptions (errors and warnings) can occur in SQL Server 2005 if a query has an existing unsafe expression—for example:

- Arithmetic exceptions—zero-divide, overflow, and underflow
- Conversion failures—loss of precision and attempts to convert a non-numeric string to a number
- Aggregation over a set of values that are not all guaranteed to be non-null
- A query plan that is changed—because changing statistics could lead to an exception in SQL Server 2005

You can prevent these run-time exceptions by modifying the query to include conditional expressions such as `NULLIF` or `CASE`. For example, the following query fails because the `x/y` expression causes a divide-by-zero error when the expression is evaluated for `y=0`:

```
SELECT x/y FROM T INNER JOIN S ON x = a AND y > b
OPTION(HASH JOIN)
```

In the following code, the expression `NULLIF(y,0)` returns `NULL` if `y = 0`. Otherwise, the expression returns the value for `y`. The expression `x/NULL` yields `NULL` and no exception occurs. This solution lets the query execute correctly:

```
SELECT x/NULLIF(y,0) FROM T INNER JOIN S ON x = a AND y > b
OPTION(HASH JOIN)
```

Functions such as `MIN` issue a warning that a null value was eliminated if its input contained a `NULL`. If you do not want the `NULL` inputs to be processed and you do not want a warning issued, you can modify your query locally to eliminate null values.

PRACTICE Examining the Default Log Trace File

In this practice session, you use the `fn_trace_gettable` stored procedure to examine the default log trace file. If you want to extend the procedure, use the Performance and Errors And Warnings event categories in Profiler to manage the file content.

► **Practice: Using the `fn_trace_gettable` Stored Procedure to Examine the Default Log Trace File**

1. Log in to your domain at your member server by using either your domain administrator account or an account that has been added to the `sysadmin` server

role. If you are using virtual machine software, log in to your domain and connect to your member server.

2. From Programs, select Microsoft SQL Server 2005, and select SQL Server Management Studio. Connect to the database engine on your member server, specifying Windows Authentication. Specify TCP/IP and the AdventureWorks database, and then click Connect.
3. Click New Query to start the Query Editor.
4. Enter the following query:

```
SELECT * FROM fn_trace_gettable('C:\Program Files\Microsoft SQL
Server\MSSQL.1\MSSQL\LOG\log.trc', default)GO
```

5. Press F5 to run the query.
6. Examine the query output. You should see a considerable amount of information about SQL Server events—for example, the event class, the username of the account under which the event ran, the server process identifier (SPID), and the name of the application. The EventSubClass and TextData columns often contain useful information.

NOTE The default trace log might not exist

The *fn_trace_gettable* stored procedure fails if the log.trc file does not exist or is corrupt. In this case, navigate to the C:\Program Files\Microsoft SQL Server\MSSQL.1\MSSQL\LOG folder and double-click the lowest numbered file. For example, if you see Log_5.trc, Log_6.trc, and log_7.trc, double-click log_5.trc. This opens the rollover file in Profiler.

Lesson Summary

- You can open and examine the default trace logs by using Profiler, or you can query them by using the Transact-SQL *fn_trace_gettable* system function.
- To obtain relevant data from a trace file, you can examine the Errors And Warnings event category and the Performance event category in Profiler.
- Run-time exceptions (errors and warnings) can occur in SQL Server 2005 if a query has an existing unsafe expression.
- You can prevent run-time exceptions by modifying the query to include conditional expressions such as NULLIF or CASE.

Lesson Review

You can use the following questions to test your knowledge of the information in Lesson 4, “Enforcing Appropriate Stored Procedure Logging and Output.” The questions are also available on the companion CD if you prefer to review them in electronic form.

NOTE Answers

Answers to these questions and explanations of why each answer choice is correct or incorrect are located in the “Answers” section at the end of the book.

- I. You need to generate a trace file that provides you with specific information. In particular, you want information about the event classes that are produced when DML operators are executed. Which event category should you examine?
 - A. Errors And Warnings
 - B. Performance
 - C. Show Plan Statistics
 - D. Trace File Properties

Lesson 5: Troubleshooting Concurrency Issues

Users who access the same resource at the same time are said to be accessing the resource concurrently. Concurrent data access requires mechanisms to prevent adverse effects when multiple users try to modify resources that other users are actively using.

Optimistic concurrency control works to minimize reader/writer blocking. With optimistic concurrency control methods, read operations do not use read locks that block data modification operations. *Pessimistic concurrency control* works to ensure that read operations access current data and that data being read cannot be modified. With pessimistic concurrency control methods, read operations use read locks that block data modification. The locks placed by a read operation are released when the read operation is finished.

After this lesson, you will be able to:

- Use SQL Server Locks performance counters.
 - Evaluate Locking/sec.
 - Set an isolation level.
 - Identify block culprits.
 - Identify deadlock chain culprits.
- Use partitioned indexes.
- Evaluate the Transactions/sec performance counter.
- Use Alerts to trigger the notification process.
- Use SQL Server Profiler to troubleshoot concurrency issues.

Estimated lesson time: 75 minutes

Using the SQL Server Locks Performance Counters

The SQLServer:Locks object returns information about SQL Server locks on individual resource types. You can use Systems Monitor to view these counters in real time, capture their values in a performance log, or use them to trigger an alert. Locks are held on SQL Server resources to prevent concurrent use of resources by different transactions. Minimizing locks increases concurrency, which can improve performance. You can monitor multiple instances of the Locks object at the same time, with each instance representing a lock on a resource type. The SQLServer:Locks object provides the following counters:

- **Average Wait Time (ms)** Average amount of wait time (in milliseconds) for each lock request that resulted in a wait

- **Lock Requests/sec** Number of new locks and lock conversions per second requested from the lock manager
- **Lock Timeouts (timeout > 0)/sec** Number of lock requests per second that timed out, but excluding requests for NOWAIT locks
- **Lock Timeouts/sec** Number of lock requests per second that timed out, including requests for NOWAIT locks
- **Lock Wait Time (ms)** Total wait time (in milliseconds) for locks in the last second
- **Lock Waits/sec** Number of lock requests per second that required the caller to wait
- **Number of Deadlocks/sec** Number of lock requests per second that resulted in a deadlock

Evaluating Locking/sec

The SQL Server:Locks performance counters tell you the type of locks and how many times per second SQL Server applies them. If you have multiple processes running against a database or a large number of users submitting ad hoc queries, locks are essential to implement concurrency and preserve data integrity. You should, however, obtain baseline readings. If the volume of locks per second is increasing, you might have resource problems. In particular, if the number of deadlocks per second is increasing, application and query performance could be negatively affected and you need to identify deadlock culprits.

MORE INFO SQLServer:Locks

For more information about the SQLServer:Locks object and resources that can be locked, search Books Online for "SQL Server, Locks Object" or access [msdn2.microsoft.com/en-us/library/ms190216\(SQL.90,d=ide\).aspx](https://msdn2.microsoft.com/en-us/library/ms190216(SQL.90,d=ide).aspx).

SQL Server:Access Methods Object

In addition to the counters provided by the SQLServer:Locks object, you should monitor the Full Scans/sec counter provided by the SQL Server:Access Methods object and the SQL Server:SQL Statistics Batch Requests/sec counter. These counters have no absolute critical values, but when you measure them against a baseline, they give you an indication of how a database application is coping with both predefined and ad hoc queries. You can then use the values in the SQL Server:Locks counters to determine whether performance is affected by locking issues.

Setting an Isolation Level

The isolation level at which a transaction (application or query) runs determines how sensitive the transaction is to changes that other users' transactions make, and it determines how long the transaction must hold locks to protect against these changes. Read Committed is the default isolation level for the SQL Server 2005 database engine. This isolation level specifies that statements cannot read data that has been modified but not committed by other transactions. This prevents dirty reads.

Dirty Reads

Transactions running at the Read Uncommitted level do not issue shared locks to prevent other transactions from modifying data read by the current transaction. Read Uncommitted transactions are also not blocked by exclusive locks that prevent the current transaction from reading rows that have been modified but not committed by other transactions. When you enable the Read Uncommitted option, it is possible to read uncommitted modifications, which are called *dirty reads*. Values in the data can be changed and rows can appear or disappear in the data set before the end of the transaction. This option has the same effect as setting NOLOCK on all tables in all SELECT statements in a transaction. Read Uncommitted is the least restrictive of the isolation levels.

If a transaction needs to operate at a different isolation level, you can set the isolation level by using the Transact-SQL SET TRANSACTION ISOLATION LEVEL statement. The statement has five arguments:

- READ UNCOMMITTED
- READ COMMITTED
- REPEATABLE READ
- SNAPSHOT
- SERIALIZABLE

Real World

Ian Mclean

The implications of each isolation level are complex, and I have succeeded in reaching a hoary old age without memorizing this information. Check out “SET TRANSACTION ISOLATION LEVEL” in Books Online and you’ll see what I mean. It is sufficient to know that the higher the isolation level, the more locks that are used, and the longer transactions take to complete. If, for example, a SQL Server 2005 server is running short of memory to manage locks, or if the Business Plan requires that a particular dataset always needs to be available immediately even if other applications are running and might be modifying the data at the time (that is, dirty reads are acceptable), consider setting the isolation level to Read Uncommitted.

When you specify the isolation level, the locking behavior for all queries and DML statements in the SQL Server session operates at that isolation level. The isolation level remains in effect until you terminate the session or until you set the isolation level to another level. The following example sets the Read Uncommitted isolation level and commits a transaction at that isolation level:

```
USE AdventureWorks;
GO
SET TRANSACTION ISOLATION LEVEL READ UNCOMMITTED;
GO
BEGIN TRANSACTION;
GO
SELECT EmployeeID
FROM HumanResources.Employee;
GO
COMMIT TRANSACTION;
GO
```

CAUTION Table-level hints and plan guides

You can override the isolation level for individual queries or DML statements by specifying a *table-level hint* in the FROM clause of a statement. This does not affect other statements in the session. The hints override any execution plan the query optimizer might select for a query. However, because the SQL Server 2005 query optimizer typically selects the best execution plan for a query, Microsoft recommends that table hints (also joint hints and query hints) be used only as a last resort by experienced developers and DBAs. If a table hint in a stored procedure or an application is causing problems, and you cannot alter the procedure or application, you can use a *plan guide* to solve the problem. You use plan guides to apply query hints to queries in deployed applications when you cannot or do not want to change the application directly.

Exam Tip In light of the advice Microsoft gives about table-level hints, it is unlikely that using such a hint will be the correct solution to a problem posed in the exam. You should, however, know that you can use plan guides to apply a query hint or stop an existing hint from being applied.

If you issue `SET TRANSACTION ISOLATION LEVEL` in a stored procedure, trigger, user-defined function, or user-defined type, when the object returns control, the isolation level is reset to the level in effect when the object was invoked. For example, if you set `Repeatable Read` in a batch, and the batch then calls a stored procedure that sets the isolation level to `Serializable`, the isolation level setting reverts to `Repeatable Read` when the stored procedure returns control to the batch. `SET TRANSACTION ISOLATION LEVEL` takes effect at execute or run time, and not at parse time.

Identifying Block and Deadlock Chain Culprits

Profiler enables you to record events as they occur in an instance of the database engine. The recorded events are instances of the event classes in the trace definition. In Profiler, event classes and their event categories are available on the Events Selection tab of the Trace File Properties dialog box. You can use the event classes in the Locks event category to monitor locking activity and to identify the procedures and queries that are causing blocks and deadlocks. These event classes can help you investigate locking problems caused by multiple users reading and modifying data concurrently:

- **Deadlock Graph** Provides an XML description of a deadlock.
- **Lock:Acquired** Indicates that a lock has been acquired on a resource, such as a row in a table.
- **Lock:Cancel** Tracks requests for locks that were canceled before the lock was acquired (for example, to prevent a deadlock).
- **Lock:Deadlock Chain** Monitors when deadlock conditions occur and which objects are involved.
- **Lock:Deadlock** Tracks when a transaction has requested a lock on a resource already locked by another transaction, resulting in a deadlock.
- **Lock:Escalation** Indicates that a finer-grained lock has been converted to a coarser-grained lock. Locking at a smaller granularity, such as rows, increases concurrency but has a higher overhead because more locks must be held if many rows are locked. Locking at a larger granularity, such as tables, decreases concurrency because locking an entire table restricts access to any part of the table by other transactions.
- **Lock:Released** Tracks when a lock is released.

- **Lock:Timeout (timeout > 0)** Tracks when lock requests cannot be completed because another transaction has a blocking lock on the requested resource. This event occurs only in situations where the lock time-out value is greater than zero.
- **Lock:Timeout** Tracks when lock requests cannot be completed because another transaction has a blocking lock on the requested resource.

A Profiler trace that uses event classes in the Locks event category indicates when blocking locks and deadlocks occur. You can then use the *sp_who* stored procedure to obtain information about current users and processes. The *sp_who* stored procedure returns, for example, the login name of the user, the hostname of the computer used for the process, the system process identity (SPID) of the process, and the SPIDs of any blocking processes.

CAUTION Capturing Locks event classes can be resource intensive.

Because the database engine typically processes a large number of locks, capturing the Locks event classes during a trace can incur significant overhead and result in large trace files or tables.

Deadlock Chain Event Class The Lock:Deadlock Chain event class is produced for each participant in a deadlock. You can use this event class to monitor when deadlock conditions occur and to determine whether deadlocks are significantly affecting the performance of your application and which objects are involved. You can examine the application code that modifies these objects to determine whether changes to minimize deadlocks can be made. This event class returns a large amount of information in its data columns, including the security identifier (SID) of the user, the type of lock, and the owner (transaction, cursor session, and so on) of the process.

MORE INFO Deadlock Chain event class

For more information about the Lock:Deadlock Chain event class, search for “Lock:Deadlock Chain Event Class” in Books Online or access [msdn2.microsoft.com/en-us/library/ms189554\(d=ide\).aspx](https://msdn2.microsoft.com/en-us/library/ms189554(d=ide).aspx).

Evaluating the Transactions/sec Performance Counter

A transaction is a sequence of operations, performed as a single logical unit of work, that must exhibit four properties: the atomicity, consistency, isolation, and durability (ACID) properties, which are defined as follows:

- **Atomicity** Either all the data modifications the transaction specifies are performed or none of them is performed.

- **Consistency** When completed, a transaction must leave all data in a consistent state. In a relational database, all rules must be applied to the transaction's modifications to maintain all data integrity.
- **Isolation** Modifications made by concurrent transactions must be isolated from the modifications made by any other concurrent transactions. A transaction either recognizes data in the state it was in before another concurrent transaction modified it or recognizes the data after the second transaction has completed, but it does not recognize an intermediate state.
- **Durability** After a transaction has completed, its effects are permanently in place in the system. The modifications persist even in the event of a system failure.

The SQL Server:Databases object provides the Transactions/sec counter. This counter records only transactions that change data, or explicit transactions. Queries that do not change data are implicit—but nevertheless, they are transactions. In spite of this limitation, the Transactions/sec counter is useful for determining whether the number of transactions that run against a database has increased substantially.

If the long-term average is (say) 20 transactions per second and you suddenly see readings of 200 transactions per second or more, the problem could be a faulty application, but more likely the server hardware can no longer cope with server activity. The Transactions/sec counter reading indicates how active your SQL Server 2005 system is, rather than any specific fault. A higher value indicates more activity is occurring.

Using the Batch Requests/sec Counter

The SQL Server:SQL Statistics object provides counters to monitor compilation and the type of requests sent to an instance of SQL Server. Monitoring the number of query compilations and recompilations as well as the number of batches received by an instance of SQL Server gives you an indication of how quickly SQL Server is processing user queries and how effectively the query optimizer is processing the queries.

The Transactions/sec counter does not measure activity unless it is inside an explicit transaction. However, the SQL Server:SQL Statistics: Batch Requests/sec counter measures all batches you send to the server even if they do not participate in a transaction. SQL Server Books Online identifies the latter counter as a good indicator of throughput, although Transactions/sec is the counter that many people use in the field. SQL Server Agent alerts, however, are typically triggered on Batch Requests/sec rather than Transactions/sec.

Using Alerts to Trigger the Notification Process

An event notification is a database object that executes in response to DDL statements and trace events. When a notification executes, it sends an XML-formatted message to a Service Broker service. You can use event notifications to react to changes related to database objects. If you take advantage of the Service Broker queuing and delivery support, event notifications can be a powerful tool.

Service Broker

Service Broker helps developers build asynchronous, loosely coupled applications in which independent components work together to accomplish a task. These application components exchange messages that contain the information required to complete the task. Application development is beyond the scope of this book or the 70-444 exam, but if you want to learn more, search for “What Does Service Broker Do?” in Books Online.

An event notification responds to DDL statements and certain SQL Server trace events and sends an XML message to the Service Broker service. However, the service can be designed to activate a stored procedure that processes the message. Rollback is not supported, and event notification names are scoped by the server, database, assembly, or a specific object within a database.

Event notifications execute in response to a variety of Transact-SQL DDL statements and SQL Trace events by sending information about these events to a Service Broker service, and they can be used to log and review changes or activity occurring on the database. They run asynchronously, outside the scope of a transaction, whether the transaction commits or not.

Creating an Alert to Trigger a Notification

A job is a series of steps, performed sequentially by SQL Server Agent, that contain actions in the sequence that you want to execute them. Jobs can, for example, import data on a regular basis or back up a database. They can perform a wide range of activities, including running Transact-SQL scripts, command prompt applications, Microsoft ActiveX scripts, Integration Services packages, Analysis Services commands and queries, or Replication tasks.

MORE INFO **Creating jobs**

For more information, search for “Creating Jobs” in Books Online or access [msdn2.microsoft.com/en-us/library/ms186273\(d=ide\).aspx](https://msdn2.microsoft.com/en-us/library/ms186273(d=ide).aspx).

You can create a job or edit an existing job by accessing the Jobs folder within the SQL Server Agent object in SSMS. A job can automatically notify users of job status by generating alerts. You can configure this function by accessing the Alerts page of the job you want to configure, and either selecting a current alert or clicking Add to display the New Alert dialog box. You can also access a job’s Notification page and notify operators (typically DBAs) by e-mail, pager, or Net Send messages.

Responding to SQL Server Errors

The Alert object represents a single SQL Server Agent alert. Alerts respond to either specific SQL Server error messages or SQL Server errors of a specified severity. You can use the Alert object to create and manage SQL Server Agent alerts in the following ways:

- Creating an alert to respond to a specific SQL Server error
- Changing the properties of an existing alert to modify its behavior
- Changing the notified operators on an instance of the error condition

SQL Server does not allow you to create more than one alert on any given error condition or severity level. You can define more than one alert on a specific message identifier; however, you must limit the scope for each alert you define by associating the alert with a specific database. SQL Server alerts are enabled by default. You must, however, assign operators to the alert by using the AddNotification method of the Alert or Operator object.

MORE INFO **The AddNotification method**

For more information, search for “Alert.AddNotification Method” in Books Online or access [msdn2.microsoft.com/en-us/library/microsoft.sqlserver.management.smo.agent.alert.addnotification\(d=ide\).aspx](https://msdn2.microsoft.com/en-us/library/microsoft.sqlserver.management.smo.agent.alert.addnotification(d=ide).aspx).

To create an alert, you need to create an Alert object, set the Name property, set the response type for the alert by setting the value of the Severity property or the MessageID property as well as any optional properties you require. For example, you can set the DatabaseName property to limit the alert’s action to a specific database, or use

the `AddNotification` method to add operators to the alert. You can then add the `Alert` object to the `Alerts` collection of a connected `JobServer` object, which represents the Microsoft SQL Server Agent subsystem and the `msdb` database.

Using SQL Server Profiler to Troubleshoot Concurrency Issues

SQL Server 2005 lets you track locks being acquired and released in events; however, for almost any level of concurrent usage, events that acquire and release locks occur frequently. To troubleshoot concurrency issues, you should not attempt to capture and analyze all lock events. Instead, you should use Profiler to find deadlocks and lock timeouts. You need to monitor the following events classes:

- `Lock:Deadlock`
- `Lock:Deadlock Chain`
- `Deadlock Graph`
- `Lock:Timeout`

`Lock:Deadlock` and `Lock:Deadlock Chain` help track down the statements that are causing deadlocking issues. Database tables and views should always be accessed in the same order. However, where multiple application developers are working with the same database or multiple users are running ad hoc queries, deadlock chains can occur. By creating a trace that captures the `Lock:Deadlock` and `Lock:Deadlock Chain` events, you can detect conflicts and take the appropriate action. The `Deadlock Graph` event class provides an XML description of a deadlock, and it provides (arguably) the best method of discovering exactly why deadlocks are occurring. This event class occurs simultaneously with the `Lock:Deadlock` event class.

IMPORTANT `Lock:Timeout`

By default, SQL Server does not register a lock timeout unless the client application sets one. An application can specify a lock timeout by `SET LOCK_TIMEOUT`. Otherwise, the application will wait indefinitely for the locks to clear. If you set a lock timeout and the threshold is reached, you can capture the `Lock:Timeout` event class in the Profiler trace.

PRACTICE Saving a Deadlock Graph

In this practice session, you configure Profiler to save a deadlock graph as an XML file. If you want to extend the practice, create a deadlock event that can be captured in the file. You can do this by having two concurrent connections to a database that contains

two tables (say, Table A and Table B). On the first connection, create a query that updates Table A and queries Table B. On the second connection, create a query that updates Table B and queries Table A. End both queries with a rollback. Run both queries simultaneously.

► **Practice: Configuring Profiler to Save a Deadlock Graph**

To configure Profiler to save a deadlock graph as an XML file, complete the following steps:

1. Log in to your domain at your member server by using either your domain administrator account or an account that has been added to the sysadmin server role. If you are using virtual machine software, log in to your domain and connect to your member server.
2. From the Programs (or All Programs) menu, choose Microsoft SQL Server 2005, choose Performance Tools, and then choose SQL Server Profiler. On the File menu, choose New Trace. On the Login tab of the Connect To Server dialog box, specify a connection to the Database Engine on your member server using Windows Authentication (the default). On the Connection Properties tab, specify the AdventureWorks database and the TCP/IP protocol. Click Connect.
3. Type **DeadlockTrace** in the Trace Name text box.
4. In the Use The Template drop-down list, verify that Standard is selected.
5. Select the Save To File check box to capture the trace to a file. Select the folder you created earlier in this chapter for saving Profiler traces (or create a folder if you have not already done so), and click Save. Verify that the maximum file size is 5 MB, and ensure that the Enable File Rollover check box is selected.
6. Select the Enable Trace Stop Time check box, and specify a stop date and time.
7. On the Events Selection tab, expand the Locks event category in the Events data column, and then select the Deadlock Graph check box as shown in Figure 2-14. If the Locks event category is not available, select the Show All Events check box to display it.
8. The Events Extraction Settings tab is added to the Trace Properties dialog box. On this tab, click Save Deadlock XML Events Separately.
9. In the Save As dialog box, specify Deadlock as the name of the file in which to store the deadlock graph events. Click Save.

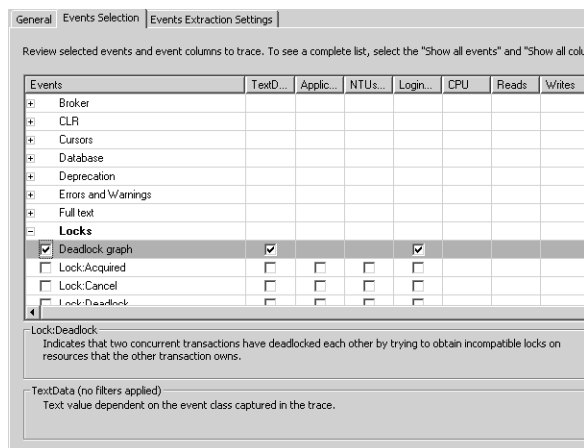


Figure 2-14 Selecting Deadlock Graph.

10. On the Events Extraction Settings tab, verify that All Deadlock XML Batches In A Single File is selected. (Optionally, you can choose to create a new XML file for each deadlock graph by selecting Each Deadlock XML Batch In A Distinct File.)
11. Click Run to save the graph in the file you created for this purpose. An XML graph will not be created until you run a query or an application against the database that causes a deadlock.

MORE INFO Opening a deadlock file

After you have saved the deadlock file, you can open it in SSMS. For more information, refer to "How to: Open, View, and Print a Deadlock File (SQL Server Management Studio)" at [msdn2.microsoft.com/en-us/library/ms187021\(SQL.90,d=ide\).aspx](https://msdn2.microsoft.com/en-us/library/ms187021(SQL.90,d=ide).aspx).

Lesson Summary

- The SQLServer:Locks object returns information about SQL Server locks on individual resource types.
- The isolation level at which a transaction (application or query) runs determines how sensitive the transaction is to changes other users' transactions make, and how long the transaction must hold locks to protect against these changes.
- You can use the event classes in the Locks event category in Profiler to monitor locking activity and to identify the procedures and queries that are causing blocks and deadlocks.
- You can use the Lock:Deadlock Chain event class to monitor when deadlock conditions occur and to determine whether deadlocks are significantly affecting the performance of your application and which objects are involved.

- You can use the SQL Server: Databases: Transactions/sec and SQL Server:SQL Statistics: Batch Requests/sec counters to determine whether the number of transactions run against a database has increased substantially.
- You can create an alert on a job to trigger a notification and configure the notification by accessing the Management folder in Management Studio on a server running SQL Server Agent.
- The Deadlock Graph event class in Profiler provides an XML description of a deadlock, and it provides an efficient method of discovering exactly why deadlocks occur.

Lesson Review

You can use the following questions to test your knowledge of the information in Lesson 5, “Troubleshooting Concurrency Issues.” The questions are also available on the companion CD if you prefer to review them in electronic form.

NOTE Answers

Answers to these questions and explanations of why each answer choice is correct or incorrect are located in the “Answers” section at the end of the book.

1. You receive alerts reporting that several transactions on your SQL Server 2005 database have terminated as a result of a deadlock error. You need to find out the causes of the deadlocks by using the least administrative effort. What should you do?
 - A. Use the *sys.dm_tran_locks* DMV.
 - B. Create a performance log to record the information from the Number of Deadlocks/sec counter in the SQL Locks object. Use System Monitor to view the log.
 - C. Copy the suspected application code into Query Editor. Run the DTA directly from Management Studio, and implement the recommendations.
 - D. Run SQL Server Profiler, and create a trace with the Deadlock Graph event class. Examine the resulting XML file.
2. You are a DBA for Northwind Traders. During business hours, Northwind employees run ad hoc and predefined queries against a read-only database. The database is updated at night. You need to monitor the database performance on

Northwind's SQL Server 2005 servers during the day. What counter should you monitor?

- A. The Lock Waits/sec performance counter
 - B. The Full Scans/sec performance counter
 - C. The Number of Deadlocks/sec performance counter
 - D. The Transactions/sec performance counter
3. You are the DBA for a manufacturing company. Managers frequently run a SQL Server 2005 database application that creates an inventory report. They need to obtain the latest inventory report quickly. The application runs queries against tables in the StockControl database that are frequently updated. Managers need to obtain an inventory report even if other applications are currently performing updates. How do you enable them to do so?
- A. Change the transaction isolation level for the inventory report transaction to Read Committed.
 - B. Create an indexed view that contains all the columns that the inventory report transaction accesses.
 - C. Change the transaction isolation level for the report transaction to Read Uncommitted.
 - D. Change the transaction isolation level for the report transaction to Repeatable Read.

Chapter Review

To further practice and reinforce the skills you learned in this chapter, you can complete the following tasks:

- Review the chapter summary.
- Review the list of key terms introduced in this chapter.
- Complete the case scenario. This scenario sets up a real-world situation involving the topics of this chapter and asks you to create a solution.
- Complete the suggested practices.
- Take a practice test.

Chapter Summary

- SQL Server 2005 provides a wide variety of powerful tools to diagnose poorly performing queries. However, often the best procedure is to list the query in Query Editor and examine it manually.
- Query plans can be displayed as Profiler traces, as graphical execution plans, or by using DMVs, which can also provide CPU and disk I/O usage statistics.
- You can reduce the number of table scans by modifying your queries and by creating indexes. You can create clustered, unclustered, and covering indexes and indexed views.
- Trace logs record SQL Server events generated by stored procedures or ad hoc queries. You can open and examine trace logs by using Profiler or a SQL query.
- You can use the event classes in the Locks event category in Profiler to monitor locking activity and to identify the procedures and queries that are causing blocks and deadlocks. You can control the degree of locking by setting the isolation level at which transactions operate.

Key Terms

Do you know what these key terms mean? You can check your answers by looking up the terms in the glossary at the end of the book.

- ad hoc query
- BLOB

- clustered index
- concurrency
- event class
- fragmentation
- handle
- heap
- index
- indexed view
- isolation level
- nonclustered index
- partitioning

Case Scenario

In the following case scenario, you apply what you've learned in this chapter. You can find answers to these questions in the “Answers” section at the end of this book.

Case Scenario: Dealing with Compatibility Problems and Fragmented Indexes

You are the senior DBA at Coho Vineyard. You administer four SQL Server 2005 member servers that contain the Production, Stock, Sales, Accounts, and Human_Resources databases. Coho Winery is a partner company, and you are required to run database applications that were developed for Coho Winery. Sometimes these can cause compatibility problems, but you cannot alter the code. Your servers are all due for replacement in the next six months, and no budget is available for upgrades in the meantime.

1. Coho Vineyard employees run an application against the Production.Merlot table. They report it is performing slowly. You discover that the server that holds this database frequently runs out of memory to manage locks. Transferring the database to another server is not an option. The application designer informs you that dirty reads are acceptable. What should you do?
2. The Sales.United_Kingdom table is very write-intensive, and queries frequently run against it. Both the clustered and nonclustered indexes on this table become fragmented shortly after being rebuilt. Both leaf and intermediate level pages split

frequently. There is sufficient disk space available to fully optimize the indexes. You need to optimize the nonclustered index on the `Monthly_Revenue` column of the `Sales.United_Kingdom` table. Which two actions should you perform?

3. Members of Coho Vineyard's financial staff run an application that was developed for Coho Winery. The application executes a query that uses an index query hint. The index query hint is not suitable for your environment, and you need to force the application to use a different query execution plan. How can you do this?

Suggested Practices

To successfully master the exam objectives presented in this chapter, complete the following tasks.

Identify Poorly Performing Queries

You can use several tools to identify poorly performing queries. The ability to choose the appropriate tool for a particular problem is developed through experience and practice. You should carry out both of the following practices:

- **Practice 1: Revisit Profiler, Management Studio, and the DTA.** You used these tools in the previous chapter. This chapter describes the specific ways the tools can be used to identify poorly performing queries.
- **Practice 2: Use stored procedures to set up a SQL trace.** Investigate the use of stored procedures in Books Online, and use the appropriate procedures to create and configure a trace.

Analyze a Query Plan to Detect Inefficiencies in Query Logic

You can generate statistics for an estimated query plan before a query executes, or you can obtain actual execution plan statistics by executing the query. You should carry out all of the following practices:

- **Practice 1: Use Query Editor to display the graphical execution plan for a query.** Display both the estimated and the actual execution plans.
- **Practice 2: Analyze a query plan.** Use Transact-SQL SET Statement options and the Profiler Showplan event classes to analyze a query plan.
- **Practice 3: Obtain query plan statistics.** Use the `sys.dm_exec_query_stats` DMV to obtain statistics about a query plan.

Maintain and Optimize Indexes

Creating the appropriate indexes and indexed views is the key to efficient query execution. Carry out the following practice:

- **Practice 1: Create an index.** Create a nonclustered index on a column in a table in the AdventureWorks database—for example, in the HumanResources.Employee table.

Enforce Appropriate Stored Procedure Logging and Output

Trace logs can provide a large amount of information about the operation of stored procedures and ad hoc queries. Carry out the following practice:

- **Practice 1: Open and filter a rolling trace log in Profiler.** Double-click the lowest numbered trace log file as described in Lesson 4, and open the rolling trace. Filter the results by using the EventClass and SPID columns.

Troubleshoot Concurrency Issues

Deadlocks can result in some queries executing very slowly and others failing to complete. Carry out the following practice:

- **Practice 1: Generate an XML description of a deadlock.** Use the Deadlock Graph event class in Profiler to create an XML description of a deadlock. You can generate a deadlock by creating two simultaneous connections to a database in Management Studio, simultaneously running queries against two tables in the database, and rolling back the queries.

Take a Practice Test

The practice tests on this book's companion CD offer many options. For example, you can test yourself on just one exam objective, or you can test yourself on all the 70-444 certification exam content. You can set up the test so that it closely simulates the experience of taking a certification exam, or you can set it up in study mode so that you can look at the correct answers and explanations after you answer each question.

Practice tests

For details about all the practice test options available, see the "How to Use the Practice Tests" section in this book's Introduction.

Chapter 3

Failure Diagnosis

The first and easiest part of failure diagnosis is figuring out that something has gone wrong with the server. The second and more difficult part of failure diagnosis is determining exactly what has gone wrong. Unless you know exactly what has gone wrong with the database or the server that hosts it, your chances of repairing the fault are not very good. In general, catastrophic failures are the easiest to diagnose. A server that fails to power on at all likely has blown its power supply. Intermittent faults can be the most difficult to diagnose. Some faults seem at first only to occur without rhyme or reason. It is only when you dive into the logs and know what to look for that identifying the exact cause of the problem becomes possible.

Exam objectives in this chapter:

- Diagnose causes of failures. Failure types include database failures, physical server failures, and SQL Server service failures.

Lessons in this chapter:

- Lesson 1: Diagnosing Database Failures 157
- Lesson 2: Diagnosing Physical Server Failures 169
- Lesson 3: SQL Server Service Failures 178

Before You Begin

To complete the lessons in this chapter, you must have completed the following tasks:

- Configured a Microsoft Windows Server 2003 R2 computer with Microsoft SQL Server 2005 Enterprise Edition SP1 as detailed in the Appendix.
- Installed an updated copy of the AdventureWorks sample database as detailed in the Appendix.

No additional configuration is required for this chapter.

Real World

Orin Thomas

The key to diagnosing problems with a database is being methodical, patient, and observant. When you observe a failure, you usually have suspicions about what might have caused it. Write your suspicions down and then try to think of what evidence might exist that directly confirms or contradicts your current hypothesis. Trawling log files is a lot easier if you have something specific in mind.

Lesson 1: Diagnosing Database Failures

This lesson concentrates on the tools that you can use to diagnose database failures. Primarily, you diagnose a database failure through examining log files for specific events related to the failure. The lesson covers the different ways that you can view log files and also examines what sort of evidence you find if one of the more common database failures occurs on a server that you are responsible for managing.

After this lesson, you will be able to:

- Understand the attributes of database engine errors.
- Diagnose a Full Transaction Log.
- Diagnose when tempdb has run out of disk space.
- Diagnose when a database has run out of disk space.

Estimated lesson time: 30 minutes

Log File Viewer

The Log File Viewer allows you to view log files from a variety of sources. The default log view depends on how you open the Log File Viewer. If you open the Log File Viewer by double-clicking a log from the SQL Server Logs folder, under the Management folder in SQL Server Management Studio (SSMS), the SQL Server Logs will be in focus. If you open it by double-clicking a log from the Error Logs folder of SQL Server Agent, the SQL Server Logs view is what you will see. You can view different logs by selecting them in the Select Logs pane as shown in Figure 3-1. It is possible to view logs from Database Mail, SQL Agent, SQL Server, and Windows NT in the Log File Viewer.

The database server also writes messages to the Application event log, which you can view using Event Viewer. Although this might seem redundant given the functionality of the Log File Viewer, it allows a server administrator who does not have any rights to the database to view specific errors related to the database. It also allows you to perform diagnoses on servers that are functional only via Emergency Management Services (EMS). If enabled, EMS enables you to connect to and diagnose problems with a computer that might have no network connectivity or that has a problem with its video card.

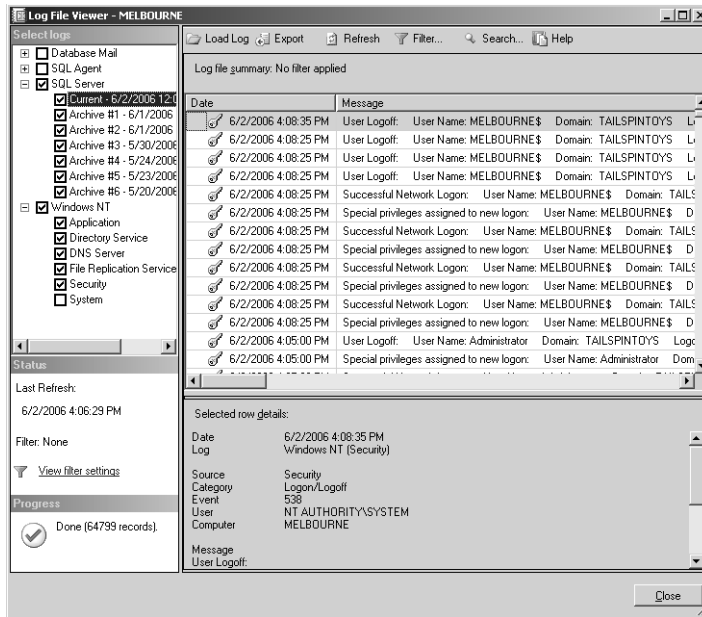


Figure 3-1 The Log File Viewer enables you to view many different logs.

NOTE Windows NT

Although the log is titled Windows NT, the title refers to the Microsoft Windows platform. You can install Microsoft SQL Server 2005 only on Windows Server 2003 or Windows 2000.

Filtering Logs

Examining a log to find information that will help you diagnose a failure can often be similar to searching for a needle in a haystack. Rather than examining one log entry after the other, you should use the filtering capacity of the Log File Viewer to narrow the events presented to you down to a manageable level. Correct use of filtering can reduce tens of thousands of log entries to just a few directly relevant ones.

The first step in filtering logs is to reduce the amount of time covered by the logs to that period when the failure that you are attempting to diagnose occurred. Next, consider filtering only for events relevant to what you are looking for. An example would be looking for a specific error source. Alternatively, you might want to search for specific text within the error message. If you specify even basic parameters such as these, you significantly reduce the number of event entries that you have to search through

while you look for that one event entry that explicitly informs you what caused the failure you are trying to diagnose.

To filter the Application Event log to show only events generated by the MSSQLSERVER source within the last day, perform the following steps:

1. Open Event Viewer from Administrative Tools.
2. Select the Application log.
3. From the View menu, choose Filter.
4. In the Event Source drop-down list of the Application Properties dialog box, select MSSQLSERVER.
5. In the From drop-down list, select Events On.
6. In the From Date drop-down list, select yesterday's date.
7. In the From Time scroll box, enter **12.00.01 AM**.
8. Click Apply, and then click OK to close the Application Properties dialog box.

NOTE Removing the filter

To remove the filter, choose Filter from the view menu, click Restore Defaults, and then click OK.

Understanding Database Engine Errors

All error messages conform to a specific format. It is important to understand the attributes of the error message that you are reading, such as the difference between severity and state. Errors that are raised by the SQL Server database engine have the attributes listed in Table 3-1.

Table 3-1 Database Engine Error Attributes

Attribute	Description
Error number	Unique error number.
Error message	Diagnostic information about the cause of the error.
Severity	The lower the number, the lower the error severity. More information on error severity can be found in Table 3-2.

Table 3-1 Database Engine Error Attributes

Attribute	Description
State	The same error number can be assigned to several different conditions. A different state number for the same error number indicates a different cause of that error.
Procedure name	The stored procedure or trigger in which the error occurred.
Line number	Which statement in a batch, stored procedure, trigger, or function generated the error.

Although the simplest way to view errors is through the Log File Viewer, it is also possible to retrieve errors using queries. The *sys.messages* catalog view contains all system and user-defined error messages. It is therefore possible to query *sys.messages* to return a list of error messages. The following query finds all error messages that have English text (1033):

```
SELECT message_id, language_id, severity, is_event_logged, text
FROM sys.messages
WHERE language_id = 1033;
```

It is of course possible to modify the query to filter for specifics, though generally using the Log File Viewer interface is easier. As mentioned in Table 3-1, when the SQL Server database engine raises an error, it assigns the error a severity level. Error severity levels range from 0 through 29. By default, only error messages with a severity level above 19 are written to the SQL Server error log. Table 3-2 describes each severity level.

Table 3-2 Error Severity Levels

Severity Level	Description
0 through 9	The database engine does not raise system errors for severity levels 0 through 9.
10	Returns informational messages that contain status information or report errors that are not severe.
11	Indicates that the given object or entity does not exist.
12	Used for queries that do not use locking.

Table 3-2 Error Severity Levels

Severity Level	Description
13	Indicates a transaction deadlock error.
14	Used for security-related errors (permission denied).
15	Indicates syntax errors in the Transact-SQL statement.
16	Indicates a general error that can be corrected by the user.
17	Indicates that a statement causes SQL Server to run out of resources (memory, locks, or disk space).
18	Indicates that a problem has occurred in the database engine software, but the statement completes execution.
19	Indicates that a nonconfigurable database engine limit has been exceeded and the current batch process has stopped.
20	Indicates that a current statement has encountered a problem but the database is unlikely to be damaged.
21	Indicates that a problem has been encountered that affects all tasks in the current database but the database is undamaged.
22	Indicates that the table or index specified in the message has been damaged by a software or hardware problem. Run DBCC CHECKDB to determine whether other objects in the database are damaged. If the problem is in the buffer cache, restarting the instance of the database engine will rectify the problem.
23	Indicates a database integrity problem caused by hardware or software. Run DBCC CHECKDB to determine the extent of the damage. If the problem is in cache, restarting an instance of the database engine will correct the problem.
24	Indicates a media failure. The database might need to be restored.

Quick Check

1. What process should you use to reduce the number of events that you need to examine in a log file?
2. What is the difference between the severity level and the state attributes of an error message?
3. At what severity level are errors written to the SQL Server log?

Quick Check Answer

1. Filtering.
2. The severity level tells you how bad the error is. The state provides details of what caused the error.
3. Errors above severity level 19 are written to the SQL Server log.

Diagnosing Common Problems Using Logs

SQL Server 2005 has comprehensive error-reporting features, but three other types of problems are more likely to turn up: tempdb is out of space, the transaction log is full, or a specific database is out of space. Just as it is useful for doctors to be able to quickly diagnose the most common ailments, it is useful for you as a database administrator to be able to use the Log File Viewer to diagnose these common problems that occur with SQL Server 2005.

tempdb Database Is Out of Space

If the tempdb database runs out of space, it causes significant problems for SQL Server 2005. You can use the *sys.dm_db_file_space_usage* dynamic management view (DMV) to monitor the use of disk space by tempdb files. The tempdb system database is available to all users connected to a particular instance of SQL Server. This system database stores user objects, internal objects, and version stores.

User Objects User objects are created by users either in the scope of a user session or within a stored procedure, a trigger, or user-defined functions. User-defined objects include the following items:

- User-defined tables and indexes
- System tables and indexes

- Global temporary tables and indexes
- Local temporary tables and indexes
- Table variables
- Tables returned in table-valued functions

Internal Objects Internal objects are created by the SQL Server database engine as it processes SQL Server statements. Objects are created and dropped within the scope of the statement. Internal objects include the following items:

- Work tables for cursor or spool operations
- Work tables for temporary large object (LOB) storage
- Work files for hash aggregate operations
- Work files for hash join operations
- Intermediate sort results for operations such as creating or rebuilding indexes, or certain GROUP BY, ORDER BY, or UNION queries

Version Stores In SQL Server 2005, there are two version stores: the common version store and an online-index-build version store. Version stores contain the following:

- Row versions generated by data modification transaction in databases using either read-committed snapshot isolation or snapshot isolation levels
- Row versions generated by online index operations, Multiple Active Result Sets (MARS), or AFTER triggers.

The SQL Server 2005 version of the tempdb system database requires significantly more space on a volume than previous versions of SQL Server. Administrators accustomed to or upgrading from previous versions often allocate less space to tempdb than it actually needs. Table 3-3 lists error messages that are written to the SQL Server log to indicate insufficient space in the tempdb database.

Table 3-3 Errors that Indicate Insufficient Space in tempdb

Error	Cause
1101 or 1105	Session must allocate space in tempdb
3959	Version store is full
3967	Version store is forced to shrink because tempdb is full
3958 or 3966	Transaction cannot find the required version record in tempdb

You can use the *sys.dm_db_session_space_usage* DMV to identify large queries, temporary tables, or table variables that are using a disproportionate amount of tempdb disk space. Tempdb is re-created every time SQL Server is started. If tempdb continues to run out of space, which is likely unless the conditions that made it run out of space in the first place are extremely unusual, you should move it to a larger volume.

MORE INFO Troubleshooting tempdb disk space

For more information about troubleshooting tempdb disk space, consult the following article on MSDN: msdn2.microsoft.com/en-us/library/ms176029.aspx.

Transaction Log Is Full

The SQL Server database engine issues a 9002 error when the transaction log becomes full. You should suspect that the transaction log is becoming full if the database is online but is in a read-only state and will not allow updates. If the transaction log fills during recovery, the database engine marks the database as RESOURCE PENDING.

Options for responding to a full transaction log include the following tasks:

- Backing up the log
- Increasing disk space or moving the log to another volume
- Increasing log file size
- Terminating long-running transactions

To determine exactly why log truncation has been prevented, you can use the *log_reuse_wait* and *log_reuse_wait_desc* columns of the *sys.database* catalog view. Factors that keep the transaction log from truncating include these:

- No checkpoint has occurred since the last log truncation.
- A data backup or restore is in progress.
- A log backup is in progress.
- A transaction is active.
- Database mirroring is paused.
- The mirror database is significantly behind the principle database.
- During transactional replications, transactions relevant to the publications are still undelivered to the distribution database.
- A database snapshot is being created.
- A log scan is occurring.

MORE INFO Transaction log troubleshooting

For more information about troubleshooting a full transaction log, consult the following articles on MSDN: msdn2.microsoft.com/en-us/library/ms175495.aspx and [msdn2.microsoft.com/en-us/library/ms345414\(d=ide\).aspx](http://msdn2.microsoft.com/en-us/library/ms345414(d=ide).aspx).

A Specific Database Is Out of Space

If a disk fills while a database is online, the database remains online but data cannot be inserted into the database. As when the tempdb database is filled, the database engine issues an 1101 or 1105 error event. If the database files are located on the same volume as the tempdb database, it is likely that both databases generate these errors. Although there are similarities between a database that is out of disk space and a full transaction log, the full transaction log does not generate the 1101 and 1105 error log events that a database running out of disk space does.

To resolve the problem, use the ALTER DATABASE command to add a file group on a separate volume to the database. It is also possible that if auto-grow is disabled, a volume might have sufficient space but the database has reached its maximum file size. You can increase the MAXSIZE value or enable auto-grow using the ALTER DATABASE statement.

Exam Tip Although it might seem that there are more possible SQL Server error messages than there are grains of sand on a beach, for this exam you should focus on the sort of error messages generated when a database runs out of disk space, the tempdb database runs out of space, and the transaction log becomes full.

MORE INFO More database engine errors

For an extensive list of errors generated by the database engine, consult the following article on MSDN: <http://msdn2.microsoft.com/en-us/library/ms365262.aspx>.

PRACTICE Filtering a Log Using the Log File Viewer

In this practice exercise, you configure the Log File Viewer to filter events so that only those generated by the source server are displayed. To perform this practice, follow these steps:

1. Open SSMS, and connect to the local database engine.
2. Expand the Management folder.

3. Expand the SQL Server Logs folder.
4. Double-click the Current log to open the log in Log File Viewer.
5. On the Log File Viewer toolbar, click Filter.
6. In the General section, click in the grid next to Source and type **Server** as shown in Figure 3-2.

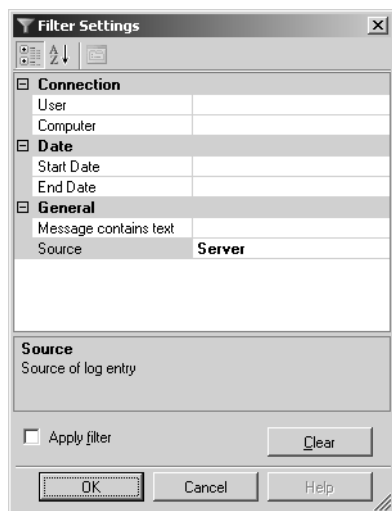


Figure 3-2 Filtering the source of the log entries.

7. Select the Apply Filter check box and then click OK.
8. Examine the log file summary. It should display only events with the source of “Server.”
9. Click Filter again. Click Clear to remove the filter and then click OK to close the Filter Settings dialog box.
10. Close the Log File Viewer.

Lesson Summary

- The Log File Viewer enables you to view logs from SQL Server, the SQL Server Agent, and Database Mail. It also allows you to view all the Windows NT event logs.
- It is possible to query the *sys.messages* catalog view to view error data.

- You can reduce the number of log entries that you have to sift through by filtering the logs.
- Database errors have the following attributes: number, message, severity, state, procedure name, and line number.
- By default, only events with a severity above 19 are written to the logs.
- If the tempdb database is out of space, users of all databases will experience problems. Error codes related to insufficient space in tempdb include 1101, 1105, 3959, 3967, 3958, and 3966.
- If the transaction log is full, it will be possible to query the database but not to update or insert information.
- If a specific database is out of space, it will not allow updates or insertions, but the situation differs from when the transaction log is full because the database engine produces error event 1101 or 1105.

Lesson Review

You can use the following questions to test your knowledge of the information in Lesson 1, “Diagnosing Database Failures.” The questions are also available on the companion CD if you prefer to review them in electronic form.

NOTE Answers

Answers to these questions and explanations of why each answer choice is correct or incorrect are located in the “Answers” section at the end of the book.

1. One of your SQL Server databases, named Mineralogy, has stopped accepting updates. You verify that the transaction log is not full and that a significant amount of disk space is still left on each of the volumes that the database uses. Other databases hosted by this SQL Server 2005 computer are functioning without problems. The error log contains several events with the code 1105. Which of the following choices most likely describes the problem?
 - A. The tempdb database has run out of disk space.
 - B. You need to enable auto-grow on the database.
 - C. You need to disable auto-grow on the database.
 - D. You need to reduce the database’s MAXSIZE.

2. The SQL Server database that you are responsible for is allowing data queries, but it refuses to accept updates. You check the error log and find an error 9002 issued by the SQL Server database engine. Transaction logs are hosted on a different volume from the other database files. Which of the following is likely to be the cause of the problem?
- A. The tempdb database is full.
 - B. The transaction log is full.
 - C. The volume hosting the database files has no free space.
 - D. The version store is full.

Lesson 2: Diagnosing Physical Server Failures

Not all problems are caused by a failure of the SQL Server 2005 software. A physical server failure is the failure of one of the server's hardware components. Hardware failures can be anything from a disk drive failing to a server catching on fire. In this lesson, we examine the hardware failures that are most likely to affect your SQL Server 2005 database.

Real World

Orin Thomas

RAM is usually the last thing you think of when trying to resolve a problem. If your computer passes its POST test, you usually figure that the source of the problem must be somewhere else. What most people don't realize is that the POST test performs only a basic RAM check and that problems that can cause a server to "bluescreen" won't necessarily be picked up by the POST RAM check. After having worked for 18 months without a problem, a server that I managed began to bluescreen. The frequency of the fault seemed random, usually once every 24 hours. The bluescreen message mentioned removing any new hardware that had been installed or any new drivers that had been updated. No new hardware had been added for almost a year, and the drivers had not been updated for a similar length of time. The only updates the computer had were the regular security updates. These updates hadn't caused problems on servers with identical hardware and software configurations, so I didn't think that was the problem.

After pulling out various components such as network cards and the graphics adapter, but still having the computer bluescreen randomly once every 24 hours, I finally rebooted into a memory diagnostic application. This application performed a battery of tests on the server's RAM. After 45 minutes of watching the text-only application successfully perform almost all its tests, I was becoming convinced I'd hit another dead end in my search for the cause of the problem. The final test, however, produced a series of errors indicating a problem with one of the sticks of RAM. I removed the troublesome stick and ran the battery of tests again, receiving no errors. I replaced the stick with a new one and restarted the server. From then on, the server functioned perfectly.

After this lesson, you will be able to:

- Diagnose disk drive failures.
- Diagnose RAID failures.
- Diagnose network card failures.
- Diagnose RAM and processor failures.

Estimated lesson time: 30 minutes

Diagnosing Volumes and Disks

You can separate diagnoses regarding volumes and disks into two basic categories: problems with the amount of storage space, and problems with the disk drive. Diagnosing a volume that has reached capacity is relatively simple. Just open My Computer and check how much free space is left. Unfortunately, because this technique is so simple, many administrators forget to regularly check how much free space is left on volumes. Disk drive failures are usually quite unexpected. How to diagnose and respond to these failures constitutes the bulk of this lesson.

Diagnosing Disk Problems

When a disk in a database server goes offline, you'll know about it pretty quickly. Files can't be accessed, and a quick check of My Computer reveals that the volume you've stored your database on is no longer present. The primary place you should go to diagnose disk and volume problems is the Disk Management folder of the Computer Management console. This console is shown in Figure 3-3. You can access this console by right-clicking My Computer and choosing Manage from the context menu.

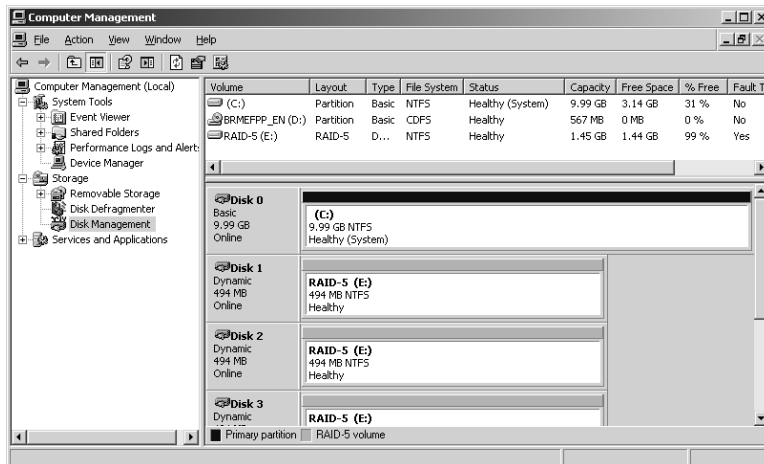


Figure 3-3 The Disk Management folder of the Computer Management console.

The top right-hand part of the console displays the currently active volumes on the server. The bottom right-hand part of the console displays the active and inactive disk drives that are connected to the server. Although some exam questions might mention lengthening disk queue counters and log entries, Disk Management provides a simple and quick way of determining if there is a problem with the disk or volume on your SQL Server 2005 computer.

NOTE Disks and volumes

It is always important to remember the terminology. In day-to-day conversation with other administrators, we often blur the terms *disk* and *volume*, but when you are looking at technical documentation, remember that the terms relate to distinctly different concepts. A disk is a physical storage device. A disk can also be used to refer to a hardware RAID array (one that is not managed by Windows). A volume is a logical partition of a disk. One disk can contain multiple volumes. It is also possible for a single volume to span multiple disks, either through spanning, striping, or RAID.

In the bottom right-hand pane, Disk Management displays the following status descriptions under each disk icon when there is a problem with a disk or the disk is not recognized:

- **Foreign** The foreign status is set when a dynamic disk from another computer is installed in a new computer. To resolve this, right-click the disk and select Import Foreign Disk.
- **Missing** This status is displayed when a dynamic disk is corrupted or disconnected. After the disk is repaired or reconnected, right-click the missing disk and choose Reactivate Disk.
- **Not Initialized** This status indicates that a disk does not contain a valid signature. To remedy this, right-click the disk and select Initialize Disk.
- **Offline** This status indicates that a dynamic disk might be corrupted or is only intermittently available. An error icon appears on the offline disk. Only dynamic disks can display the Offline status. It is possible to bring an Offline disk back online by right-clicking it and selecting Reactivate Disk.
- **Online (Errors)** Indicates that I/O errors have been detected on a region of the disk. A warning icon appears on the disk.

The Disk Management folder also displays volume status. The difference between disk status and volume status is shown in Figure 3-4. The RAID-5 volume E has a warning icon and has the volume status Failed Redundancy. One of the disks that makes up the RAID-5 volume has the error icon and is set to the status of Missing.

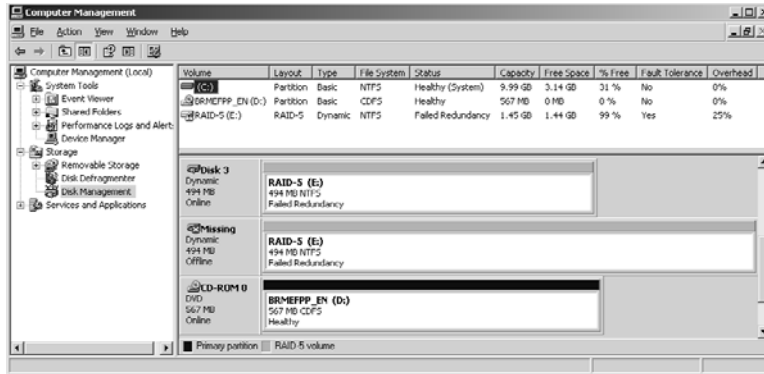


Figure 3-4 Showing both the Disk and Volume statuses when a member of a RAID-5 set is disconnected.

In the event that one of the disks in a RAID-5 volume hosting database files does fail, database performance slows dramatically. Any Disk Queue Length counter measurements taken on the volume will be much higher than the baseline. It is at this point that you should examine Disk Management to view the volume status. The following is a list of possible volume statuses displayed in Disk Management when performance has been impaired:

- Failed** This status occurs when the volume is damaged or the file system is corrupt. The failed status might indicate data loss. It is sometimes possible to return the volume to healthy status by using the Reactivate Disk command. If this does not work, you should then try using the Reactivate Volume command.
- Failed Redundancy** As shown earlier in Figure 3-4, a failed redundancy indicates that a volume that has redundancy (RAID-5, mirrors) no longer enjoys such protection. Volumes with failed redundancy can still be accessed, but performance might be significantly affected. Replacing or reactivating the missing disk will help remove this error.

MORE INFO RAID

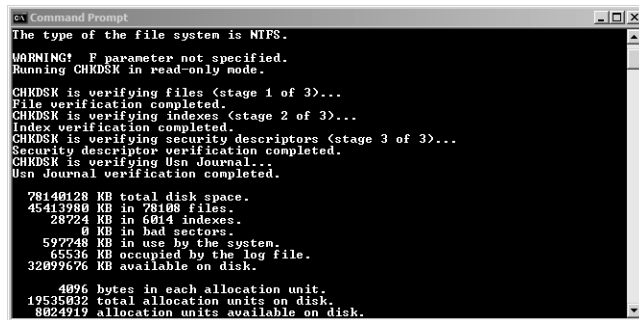
For more information about the types of fault-tolerant disk configurations supported by Windows Server 2003, consult the following article on Microsoft TechNet: technet2.microsoft.com/WindowsServer/en/Library/cb871b6c-8ce7-4eb7-9aba-52b36e31d2a11033.msp?mfr=true.

- Healthy (At Risk)** This status occurs when a dynamic volume experiences I/O errors. These errors are caused by bad sectors on the actual disk. If the errors are transient, you can use the Reactivate Disk command to return the volume to healthy status. If Reactivate Disk doesn't work, replace the disk as quickly as possible.

You should be especially cautious about continuing to use volumes that have the status Healthy (At Risk) because, like rust tends to spread, disk errors tend to multiply. The first time you might become aware of a problem is when you attempt to open, delete, or rename a file or folder that is stored on a part of the volume that has errors, and Windows Server 2003 displays a message that states the following:

```
The file or directory filename is corrupt and unreadable.  
Please run the CHKDSK utility
```

Volumes with file system errors are termed “dirty.” Unless repaired, a dirty volume will be thoroughly checked the next time the computer is rebooted. You can check a disk for errors on a volume by opening a command prompt and issuing the command CHKDSK. The output of CHKDSK is shown in Figure 3-5. If you use the /F option, CHKDSK attempts to repair any errors that it encounters on the volume.



```
Command Prompt
The type of the file system is NTFS.
WARNING! F parameter not specified.
Running CHKDSK in read-only mode.

CHKDSK is verifying files (stage 1 of 3)...
File verification completed.
CHKDSK is verifying indexes (stage 2 of 3)...
Index verification completed.
CHKDSK is verifying security descriptors (stage 3 of 3)...
Security descriptor verification completed.
CHKDSK is verifying Usn Journal...
Usn Journal verification completed.

78140128 KB total disk space.
45413980 KB in 78108 files.
28724 KB in 6014 indexes.
0 KB in bad sectors.
597748 KB in use by the system.
65536 KB occupied by the log file.
32099676 KB available on disk.

4096 bytes in each allocation unit.
19535032 total allocation units on disk.
8024919 allocation units available on disk.
```

Figure 3-5 Output of CHKDSK.

You can access a scaled-down version of CHKDSK from a volume’s properties. On some occasions, a volume might need to be dismounted to be repaired. You can accomplish this task manually using CHKDSK or, in the case of the system volume, the next time the computer is rebooted.

Exam Tip If a question mentions suddenly lengthening disk counters on a RAID volume, you can almost be certain that it is asking about a failed member in a RAID-5 set.

BEST PRACTICES Minimize the use of spanned volumes.

If possible, avoid spanning volumes across disks. The failure of one disk means that you lose all data within the spanned volume. Volumes spanned across multiple disks have a greater chance of failure than volumes located on a single disk. Many administrators make the mistake of extending a volume across disks as a way of dealing with a lack of space on an existing volume. Although this solves the problem of a lack of space on an existing volume, you should use this approach only as a stopgap measure until you can find a better solution.

Quick Check

1. Which tool enables you to quickly determine whether a disk or a volume is offline?
2. Which command-line utility should you use to check a disk for errors?
3. What is the difference between a Missing and an Offline disk?

Quick Check Answer

1. Disk Management.
2. CHKDSK.
3. Missing disks aren't connected or are corrupted. Offline disks are intermittently available. Only dynamic disks can be marked as Offline.

Diagnosing RAM and Processor Problems

RAM and processor problems are often the most difficult faults to diagnose, as these faults often tend to be intermittent rather than continuing on a regular basis. When a fault occurs with a computer's RAM or processor, Windows usually displays a STOP error. A STOP error appears as white text on a blue background. STOP errors occur only when a problem cannot be handled using Windows Server 2003 error-handling mechanisms. STOP errors provide some guidance with their output, usually suggesting that a recently installed driver or hardware device might be causing a problem. If a STOP error occurs, you should take down the error number, the error parameters and, if provided, the driver information. If there is no driver information and if you have not recently upgraded any drivers or installed any new hardware device, the possibility exists that something has gone wrong with the RAM or processor.

To resolve STOP errors, you should try the following:

- Remove any new hardware that you have installed.

CAUTION Static kills components.

Whenever you are working with the internal components of a server, remember to properly ground yourself electrically. If a static charge builds up on your body and you are not properly grounded, the static electricity could discharge onto one of the server components, causing unintended and expensive damage.

-
- Roll back any drivers that you have recently upgraded.

- Swap out existing components (such as network cards) one at a time to see whether this resolves the error. A fault on the component might be causing the STOP error.
- Test the RAM using a RAM testing utility.
- As a last resort, swap out the motherboard or processor and see whether this resolves the problem.

MORE INFO Troubleshooting STOP errors

For more information about troubleshooting STOP errors, consult the following article on Microsoft TechNet: www.microsoft.com/downloads/details.aspx?FamilyID=859637b4-85f1-4215b7d0-25f32057921c&DisplayLang=en.

Diagnosing Other Hardware Problems

Other types of hardware problems tend to be relatively simple to diagnose. A lack of picture on a monitor can be caused by a new video card driver, a broken graphics adapter, or a nonfunctional monitor. If the computer suddenly cannot connect to the network, you should check the network card to see that it is still functioning or whether there is a problem with the drop cable that runs to the patch point. Unless a server room is completely clean, the power supply—one of a server's cheapest components—might clog up with dust. The only way to stop the failure of a power supply bringing down a server is to invest in a server that has a redundant power supply.

Hardware problems are most likely to occur when you change something. Updating a SCSI driver might cause it to stop working. Inserting a new piece of hardware might cause a conflict. After you have a server functioning well, you should plan far in advance before you decide to alter that configuration. The best place to locate hardware that has failed or is suffering conflicts is in the Device Manager.

To use the Device Manager to look for failed hardware, perform the following steps:

1. Open the Control Panel.
2. Select System.
3. Click the Hardware tab.
4. Open the Device Manager.
5. From the View menu, choose Show Hidden Devices.
6. Expand each of the nodes to determine what devices are installed.
7. Any devices that have a Warning or an Error icon are problematic.

PRACTICE Using CHKDSK

In this practice, you perform a scan of a volume looking for errors. To complete this practice, perform the following steps:

1. From the Start menu, choose Run.
2. In the Run dialog box, enter **cmd.exe** and click OK to open a command prompt.
3. From the command prompt, enter the command **CHKDSK /F**.
4. If you are informed that CHKDSK cannot run because the volume is in use by another process and you are asked, “Would you like to schedule this volume to be checked the next time the system restarts?”, type **Y**.
5. Reboot the server, and allow CHKDSK to execute during the boot process.

Lesson Summary

- The Disk Management folder of the Computer Management console is the first place you should look if you suspect a problem with a disk or volume.
- A disk that has the status Missing is corrupted or disconnected.
- A disk that has the status Offline is only intermittently available.
- A disk that has the status Online (errors) has experienced I/O errors.
- A volume that has the status Failed has a corrupt file system.
- A volume that has the status Failed Redundancy still works, but it is no longer redundant.
- A volume that has the status Healthy (At Risk) is experiencing I/O errors.
- You can use CHKDSK to diagnose and repair volumes.
- RAM and processor problems usually cause STOP errors.
- The installation of new hardware or drivers can cause problems on a server and should be done only when due consideration is given to the effects the changes might have.
- Use the Device Manager to diagnose failed hardware and device conflicts.

Lesson Review

You can use the following questions to test your knowledge of the information in Lesson 2, “Diagnosing Physical Server Failures.” The questions are also available on the companion CD if you prefer to review them in electronic form.

NOTE Answers

Answers to these questions and explanations of why each answer choice is correct or incorrect are located in the "Answers" section at the end of the book.

1. You are examining disks in Disk Management and notice a warning icon on one of your server's dynamic disks. What does this warning icon indicate?
 - A. That errors have been detected on the disk
 - B. That the disk is missing
 - C. That the disk is offline
 - D. That this disk is foreign

2. One of the databases on one of the SQL Server 2005 computers that you manage has suddenly experienced a significant decrease in performance. When you open My Computer, you see that all the volumes that are supposed to be present are still available. Which of the following events are most likely to have occurred? (Choose all that apply.)
 - A. One of the disks in a RAID-5 set has failed.
 - B. One of the disks has gone offline.
 - C. The status of one of the volumes has shifted to Healthy (At Risk).
 - D. One of the volumes has failed.

Lesson 3: SQL Server Service Failures

In between potential failures of the database engine and server hardware lies the potential failure of the services that support SQL Server 2005. Services can fail for a variety of reasons—from having their startup time out during the boot process to having the password assigned to the service account expire if it is not configured correctly. This lesson deals with the tools that you can use to diagnose service failures and how you can rectify the failure of services.

After this lesson, you will be able to:

- Diagnose which SQL Server 2005 services have failed to start.
- Diagnose causes of SQL Server 2005 service failure.
- Change how SQL Server 2005 responds to failure.

Estimated lesson time: 30 minutes

SQL Server 2005 Services

There are several methods to determine which of the services that SQL Server 2005 relies on have failed to start. These include using the Services console, using the SQL Server Configuration Manager, and using logs such as the Windows System log.

Using the Services Console to Determine Status

The Services console provides an “at a glance” way of determining which services that are configured to start automatically have not. As shown in Figure 3-6, the Services console lists the service name, status, startup type, and account that the service uses to log on. A blank entry in the Status column indicates that the service has not started. If a blank entry corresponds to a Startup Type of Automatic, as is the case with the SQL Server Integration Services service in Figure 3-6, you can conclude that the service has failed to start.

From the Services console, you can edit the properties of a service and discover its dependencies. To edit the properties of a service, perform the following steps:

1. Open the Services console from the Administrative Tools menu.
2. Select the service you want to view the properties of.
3. From the Action menu, choose Properties.

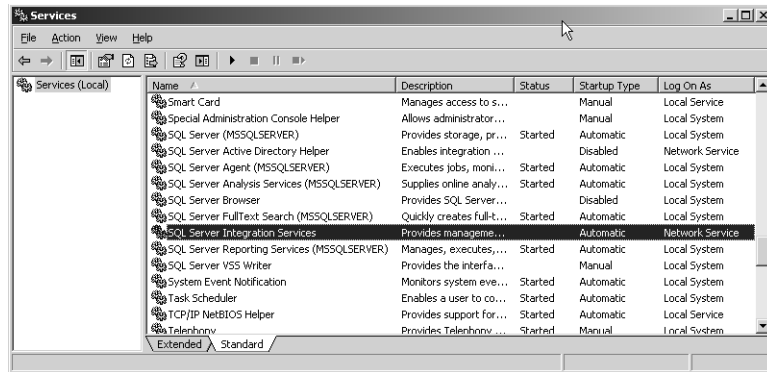


Figure 3-6 Diagnosing services using the Services console.

Figure 3-7 shows the General tab of the Properties dialog box for SQL Server Integration Services. On this tab, it is possible to change the service startup type from Automatic to Manual or to disable the service. It is also possible to start, stop, pause, or resume the service, depending on its current state. If necessary, it is also possible to enter start parameters for the service by using this tab.

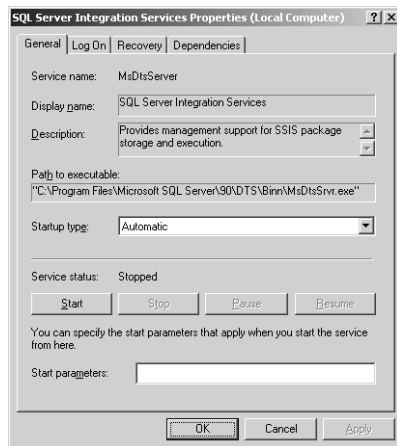


Figure 3-7 The General tab of a service's Properties dialog box.

The Log On tab of a service's Properties dialog box, shown in Figure 3-8, allows you to specify whether the service logs on using the Local System Account or another specific account. It is also possible on this tab to configure whether a service starts with a particular profile. As will be discussed later in the lesson, one of the primary causes of services failing is that the password of the account that the service runs under has expired as a result of a domain password policy.

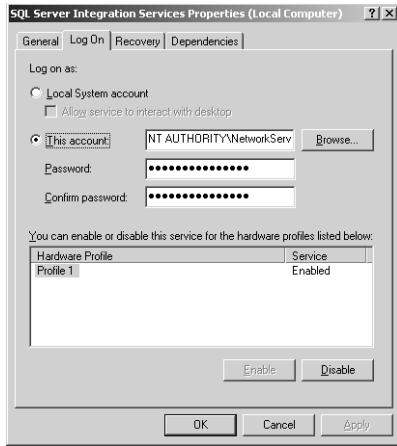


Figure 3-8 The Log On tab of a service's Properties dialog box.

The Recovery tab enables you to configure the service's behavior in the event that the service fails. Figure 3-9 shows that on the first service failure, the service is configured to restart. On the second service failure, the service is configured to restart, and on any subsequent failure the computer is restarted. The Reset Fail Count After option is set to 1 day, which means that the computer is restarted only in the event that this particular service fails three times in a 24-hour period. By default, none of these recovery options is set and a service that fails remains in that state until the administrator takes action.

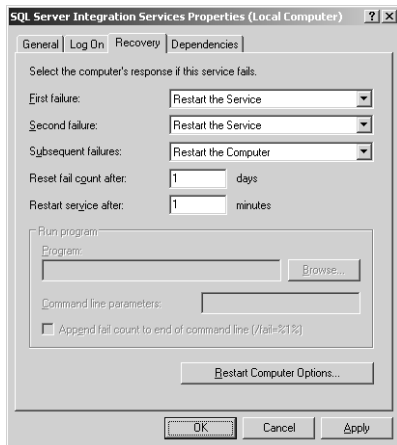


Figure 3-9 Configuring a service's recovery options.

The final tab available in a service's Properties dialog box is the Dependencies tab. You can use the Dependencies tab to determine which system components the service requires to be active to function properly. As shown in Figure 3-10, you can also determine which system components depend on this service to function properly.

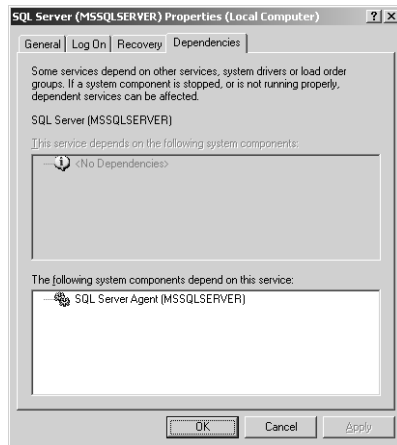


Figure 3-10 The Dependencies tab of the SQL Server service Properties dialog box displays system components that are dependent upon the service and vice versa.

Diagnosis Using SQL Server Configuration Manager

You can also use SQL Server Configuration Manager to determine which services are not running. When you select the SQL Server 2005 Services node, a list of only those services relevant to SQL Server 2005 is displayed, as shown in Figure 3-11. As is the case with the Services console, it is possible to start, stop, pause, resume, and restart services by selecting the service and choosing the appropriate item from the Action menu.

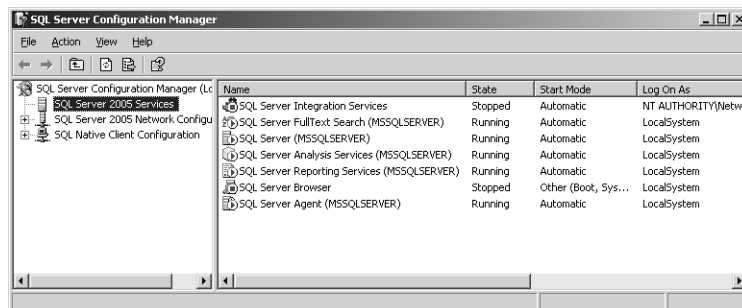


Figure 3-11 A list of SQL Server 2005 services displayed in the SQL Server Configuration Manager.

It is also possible to edit service properties using the SQL Server Configuration Manager. To do this, select the appropriate service and then choose Properties from the Action menu. As shown in Figure 3-12, the service properties available when viewing through SQL Server Configuration Manager have some similarities to the properties shown through the Services console. It is possible to specify which account the service uses to log on. It is also possible to start the service from this tab.

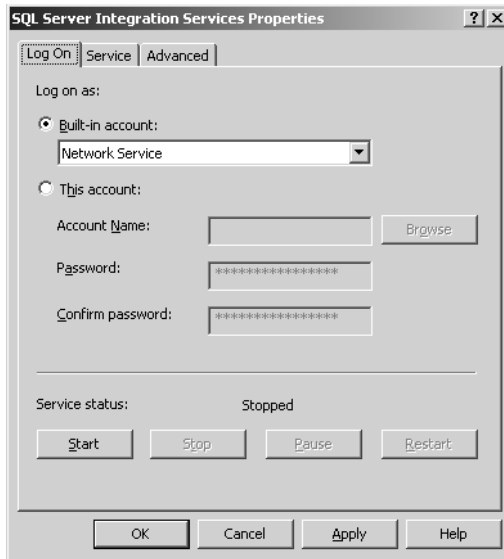


Figure 3-12 The Log On tab of a service edited through SQL Server Configuration Manager.

As shown in Figure 3-13, the Service tab of the service's Properties dialog box looks significantly different from its counterpart in the Services console. On this tab, it is possible to change the Start Mode of the service. The tab also provides other information, but it is not possible to determine service dependencies using the properties accessible from the SQL Server Configuration Manager.

Depending on which service you select, the Advanced tab can allow the configuration of a Dump Directory, Error Reporting, and Startup Parameters. The ability to configure error reporting, as shown in Figure 3-14, can be very useful for attempting to diagnose why a service might keep on failing.



Figure 3-13 The Service tab of a service edited through SQL Server Configuration Manager.

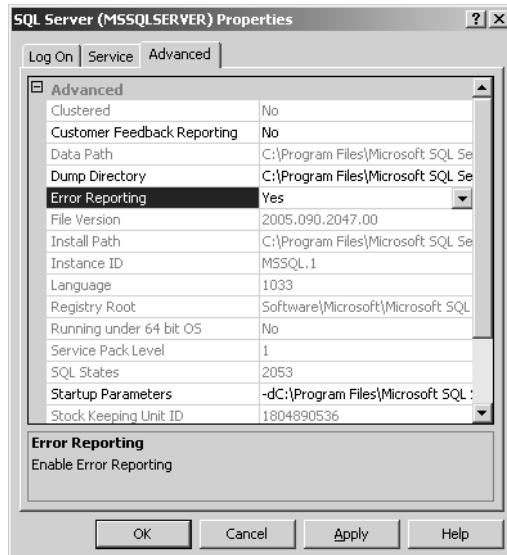


Figure 3-14 Configuring service error reporting using the SQL Server Configuration Manager.

Diagnosis Using the Windows System Log

Although both the Services console and the SQL Server Configuration Manager can tell you which services have not successfully started, they provide little information as

to why these services have not started. When a service fails to start, an event is written to the System log. You can access the System log either through Event Viewer or the appropriate node of the Log File Viewer. When using Event Viewer, you can double-click entries to read more about the reason the service failed to start, as shown in Figure 3-15. Remember to use filters so that you can reduce the list of events displayed to those that pertain directly to the service you are interested in.

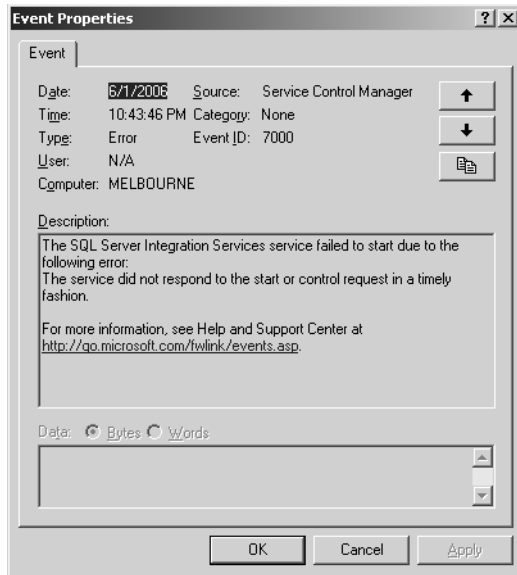


Figure 3-15 SQL Server Integration Services (SSIS) service has failed to start.

In the case shown in Figure 3-15, the SSIS service failed to start when the computer booted. The first step that the administrator should take is attempting to manually start the service. If the service fails to manually start, she should return to the logs to try to ascertain why. Services that fail to start when a computer boots can often start without a problem if manually started later. Faults like this are often transient. It is only if the fault consistently occurs that you should delve deeper.

Service Password Expiration

For security purposes, service accounts are often run under separate security accounts rather than by using the local system account. One drawback to this approach is when the security account's password properties are not correctly configured. Services that run under a security account's context fail if the domain password policy disables accounts for which the password is not changed.

The only way that you can stop a password from expiring when there is a password expiration policy is to edit the account properties and select the Password Never Expires option. Custom service accounts are almost always local accounts on member servers, though the procedure is relatively similar for domain accounts. To set the Password Never Expires option on a local computer account, perform the following steps:

1. Open the Computer Management console from the Administrative Tools menu.
2. Expand the Local Users And Groups node.
3. Select the Users folder.
4. Double-click the account used by the service.
5. On the General tab, shown in Figure 3-16, ensure that the User Cannot Change Password and Password Never Expires check boxes are selected.

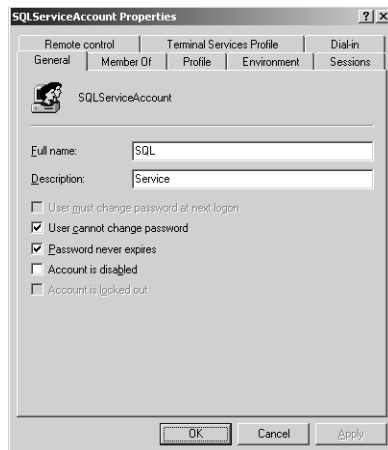


Figure 3-16 Local User Account properties.

SQL Browser Service and DAC

The dedicated administrator connection (DAC) is a special diagnostic connection that administrators can make to a SQL Server 2005 computer when other options are not available. The DAC requires that the SQL Server Browser service be active. By default, this service is disabled. When you use DAC, the SQL Server client attempts to contact the SQL Browser service to ascertain which port the DAC is listening on. If the SQL Browser service is not running, it is impossible to ascertain this information. You can enable the SQL Server Browser service either through the Services console or through the SQL Server Configuration Manager. If a server is really in trouble, it might be possible to enable the SQL Server Browser service using Emergency Management Services.

Quick Check

1. For what reasons do service account passwords often expire?
2. What tools can you use to determine whether a service is running?
3. Which tool would you use to enable error logging for a SQL Server service?

Quick Check Answer

1. Passwords expire because of domain password policy and because the accounts do not have the Password Never Expires setting enabled.
2. To determine whether a service is running, you can use the Services console, Log File Viewer, and SQL Server Configuration Manager.
3. You would use the SQL Server Configuration Manager.

The SQL Server Agent Service

The SQL Server Agent is a Microsoft Windows service that allows you to schedule the execution of administrative tasks. Scheduled administrative tasks are called *jobs*. Jobs can contain one or more tasks, such as backing up the database or executing an SSIS package. You can configure SQL Server Agent to run a job on a schedule or in response to a specific database event.

If the SQL Server Agent service fails or stops unexpectedly, the jobs that it is responsible for executing will not run. The failure of the SQL Server Agent appears within the logs, but the failure of the jobs that it is scheduled to execute does not. If you notice that backups are not being taken or that SSIS packages are not being executed, you should check the status of the SQL Server Agent service.

MORE INFO Automating administrative tasks

For more information about automating administrative tasks by using SQL Server Agent, consult the following Web site: [msdn2.microsoft.com/en-us/library/ms187061\(d=ide\).aspx](http://msdn2.microsoft.com/en-us/library/ms187061(d=ide).aspx).

PRACTICE Configuring a Service to Automatically Restart

In this practice, you alter the properties of the SQL Server Integration Services service so that it continues to automatically restart five minutes after it fails. To perform this practice, execute the following steps:

1. From the Administrative Tools menu, choose Services.
2. In the Services console, select the SQL Server Integration Services service.

3. From the Action menu, choose Properties.
4. In the SQL Server Integration Services Properties dialog box, click the Recovery tab.
5. In the First Failure drop-down list, select Restart The Service.
6. In the Second Failure drop-down list, select Restart The Service.
7. In the Subsequent Failures drop-down list, select Restart The Service.
8. Change the Restart Service After interval to 5 minutes.
9. Click OK to accept the changes.

Lesson Summary

- You can use the Services console and the SQL Server Configuration Manager to determine whether a service is running.
- By using the Services console, you can alter the account a service runs under, a service's startup type, and how the service reacts when it fails.
- You can examine a service's dependencies using the service's console.
- You can configure detailed logging for a particular service by using the SQL Server Configuration Manager.
- You can diagnose why a service failed by using the SQL Server error logs or the Windows System log.
- A common error that causes the failure of services is when the password of the account that the service runs under fails.
- The SQL Browser Service is required for remote administration via a dedicated administrator connection (DAC).
- The SQL Server Agent is responsible for running scheduled jobs. If this agent fails, backup jobs and SSIS packages will not execute.

Lesson Review

You can use the following questions to test your knowledge of the information in Lesson 3, "SQL Server Service Failures." The questions are also available on the companion CD if you prefer to review them in electronic form.

NOTE Answers

Answers to these questions and explanations of why each answer choice is correct or incorrect are located in the "Answers" section at the end of the book.

1. Which of the following tools allows you to view a service's dependencies?
 - A. Services console
 - B. SQL Server Management Studio
 - C. SQL Server Configuration Manager
 - D. SQL Server Log File Viewer
2. The SQL Server service appears to fail at random intervals. Which tool would you use to configure logging for this service?
 - A. Services console
 - B. SQL Server Management Studio
 - C. SQL Server Configuration Manager
 - D. SQL Server Log File Viewer

Chapter Review

To further practice and reinforce the skills you learned in this chapter, you can complete the following tasks:

- Review the chapter summary.
- Review the list of key terms introduced in this chapter.
- Complete the case scenarios. These scenarios set up real-world situations involving the topics of this chapter and ask you to create a solution.
- Complete the suggested practices.
- Take a practice test.

Chapter Summary

- The Log File Viewer allows you to view logs from SQL Server, the SQL Server Agent, and Database Mail. It also allows you to view all the Windows NT event logs. Searching the logs for particular error codes can help quickly diagnose problems.
- The disk management folder of the Computer Management console is the first place you should look when you suspect a problem with a disk or volume. The folder will display icons indicating problems with disks.
- CHKDSK can be used to diagnose and repair volumes, and RAM and processor problems usually cause STOP errors.
- The installation of new hardware or drivers can cause problems on a server and should be done only when due consideration is given to the effects the changes might have.
- By using the Services console, you can alter the account a service runs under, a service's startup type, and how the service reacts when it fails. It is also possible to examine a service's dependencies by using this console.
- The SQL Server Agent is responsible for running scheduled jobs. If this agent fails, backup jobs and SSIS packages will not execute.

Key Terms

Do you know what these key terms mean? You can check your answers by looking up the terms in the glossary at the end of the book.

- dedicated administrator connection (DAC)
- dependencies
- Failed Redundancy
- filter
- Log File Viewer
- Missing
- Offline
- Severity

Case Scenarios

In the following case scenarios, you will apply what you've learned in this chapter. You can find answers to these questions in the "Answers" section at the end of this book.

Case Scenario 1: Diagnosing Database Configuration Errors

Contoso's mineral exploration division has SQL Server 2005 database servers installed on all of its Antarctic exploration vehicles. You have received a radio report from one of the vehicles informing you that members of the exploration team are experiencing some problems with their database. You suspect that the problems are related to either the tempdb database running out of space, the transaction log filling, or the database that stores the mineralogical data the exploration vehicles are collecting running out of space. You need to prepare a set of questions for the vehicle crew so that they can help you diagnose the problem. In framing the questions, you need to determine what evidence they should look for.

1. What evidence would be present if the tempdb database had run out of space?
2. What evidence would be present if the transaction log was full?
3. What evidence would be present if the Mineralogical database had run out of space?

Case Scenario 2: Diagnosing Database Hardware Errors

Your contact at Tailspin Toys has called to complain that the SQL Server 2005 database server you installed appears to be having some problems. After she describes the situation to you, you suspect the following:

- One of the disk drives in the software RAID-5 array might have failed.
- Several SSIS packages that you have configured to run on the server are not executing.
- Another volume might be experiencing I/O errors.

Answer the following questions:

1. What evidence would be present if one of the disk drives in the RAID-5 array had failed?
2. What is the likely cause of the backup jobs and SSIS packages not executing?
3. What evidence would be present if there were I/O errors on a disk or volume?

Suggested Practices

To help you successfully master the exam objectives presented in this chapter, complete the following practice tasks:

Diagnose Causes of Failures

- **Practice 1: Configure a Log File Viewer filter.** Configure a filter on the current SQL Server error log to show only events that have occurred in the past 24 hours.
- **Practice 2: Examine disk and volume status.** Use Disk Management to examine the disk and volume status. If you are using Virtual Machine software such as Virtual Server 2005 R2, create a RAID array using SCSI virtual disks and then remove one of the disks. Examine the RAID array.
- **Practice 3: Modify service properties.** Use the appropriate tool to configure service logging for the SQL Server service.
- **Practice 4: Restart a service.** Use the Services console to restart the SQL Server service. Once the service is restarted, look for evidence of this restart using the Log File Viewer. In how many places is this service restart recorded?

Take a Practice Test

The practice tests on this book's companion CD offer many options. For example, you can test yourself on just one exam objective, or you can test yourself on all the 70-444 certification exam content. You can set up the test so that it closely simulates the experience of taking a certification exam, or you can set it up in study mode so that you can look at the correct answers and explanations after you answer each question.

MORE INFO Practice tests

For details about all the practice test options available, see "How to Use the Practice Tests" in this book's Introduction.

Chapter 4

Disaster Recovery

Disaster recovery is perhaps the most important part of the database administrator profession. The effectiveness of a database administrator's response to a disaster can determine whether a company remains solvent or goes out of business. Consider an online shop. If the database is not available to process orders, the online shop will have no customers. A competent database administrator (DBA) who has developed a disaster recovery plan knows exactly what steps she needs to take to get the online storefront back up as soon as possible. A database administrator who has not planned ahead might take days or, even worse, never be able to get the online storefront back into operation. This chapter examines the ways to make a Microsoft SQL Server 2005 database fault-tolerant and covers strategies that you can use to bring back a SQL Server 2005 database from catastrophic failure—from undoing the deletion of an important table to rescuing data from a corrupt backup tape.

Exam objectives in this chapter:

- Plan for fault tolerance.
- Recover from a failure of SQL Server 2005.
- Recover from a database disaster.
 - Plan a strategy.
 - Restore a database.
 - Configure logins.
 - Recover lost data.
 - Maintain server and database scripts for recoverability.
- Salvage good data from a damaged database by using restoration techniques.

Lessons in this chapter:

- Lesson 1: Planning for Fault Tolerance 195
- Lesson 2: Recovering from Failure 207
- Lesson 3: Recovering from Database Disaster 222
- Lesson 4: Salvaging Data from a Damaged Database 237

Before You Begin

To complete the lessons in this chapter, you must have completed the following tasks:

- Configured a Microsoft Windows Server 2003 R2 computer with Microsoft SQL Server 2005 Enterprise Edition SP1 as detailed in the Appendix.
- Installed an updated copy of the AdventureWorks sample database as detailed in the Appendix.

No additional configuration is required for this chapter.

Real World

Orin Thomas

When disaster hits, remember to stay calm. Some of your workmates will panic, but you need to keep your head on straight. Your sole concern is getting things running again as soon as possible. As a smart DBA, you will have developed a disaster recovery strategy and you'll know exactly what steps you need to take. You'll have a supply of replacement hard disk drives to swap out for the ones that failed, or you'll have a standby server that has been quietly idling away in the server room for the last year doing nothing except keeping mirrored copies of the databases that are critical to your organization's operation. Stay calm, and remember that you already know what can go wrong and how to resolve it—it is just a matter of putting theory into practice.

Lesson 1: Planning for Fault Tolerance

This lesson concentrates on the methods that a database administrator can employ to ensure that a SQL Server 2005 database is fault-tolerant.

After this lesson, you will be able to:

- Understand which type of RAID array is most appropriate for specific SQL Server 2005 storage needs.
- Understand the benefits and drawbacks of using failover clustering with SQL Server 2005.
- Understand the benefits of implementing database mirroring with SQL Server 2005.
- Understand the benefits of implementing log shipping with SQL Server 2005.

Estimated lesson time: 45 minutes

SQL Server and RAID

RAID is an acronym for *redundant array of inexpensive disks*. Configuring RAID is one way of protecting the computer that hosts a SQL Server 2005 instance from the failure of a hard disk drive. Software RAID levels 0, 1, 5, and 10 are commonly used with Microsoft SQL Server 2005. RAID 10 is supported only in hardware and not by the operating system. Each type of software RAID has benefits and drawbacks in terms of performance and redundancy. A good rule of thumb is that the better the redundancy, the smaller the performance benefit.

Exam Tip When Microsoft exams discuss RAID, they almost always do so in the context of software rather than hardware RAID. Software RAID is managed by Windows; hardware RAID is managed by a hardware device such as a RAID adapter. From a Windows perspective, a configured hardware RAID device appears as a single disk. If a disk in a hardware RAID array fails, you deal with it at the hardware level rather than by using the Computer Management console. Depending on the hardware RAID, the product might ship with its own drivers and software. Generally, you do not troubleshoot and restore hardware RAID devices using the Disk Management console, though you might use that console to work with volumes hosted on hardware RAID disks.

RAID 0

RAID 0 is also known as *disk striping*. When you implement RAID 0, data is divided into blocks and then spread equally over the disks that make up the RAID 0 set. RAID 0 provides the best performance, as data reads and writes aren't limited to a single disk but can be performed simultaneously across all disks on the set. To explain this

via an exaggerated example, if a 75-MB file needs to be written to a normal single disk volume and the disk drive/controller combination had a write speed of 25 MB per second, it would take three seconds to write the file. If you configure three controller/disk drive sets of the same speed as a RAID 0 set, the same file would take only one second to be written to the disk. This efficiency is gained because each controller/disk set would write a 25-MB part of the file to its part of the RAID 0 set. The drawback to a RAID 0 set is that it has no redundancy. If one of the disk drives that constitute a RAID 0 volume fails, all data on the RAID 0 volume is lost.

RAID 1

RAID 1 is also known as *disk mirroring*. A RAID 1 volume uses two separate disks, with the second disk duplicating all data stored on the first. RAID 1 does not provide any performance benefit, but it does provide a redundancy benefit. If one of the disks in the mirrored pair fails, data remains accessible because an exact copy of it exists on the other disk. Repairing a mirrored set is a matter of breaking the set, replacing the failed disk with a new one, and then building a new mirrored set.

RAID 5

RAID 5 is sometimes referred to as *disk striping with parity*. RAID 5 requires a minimum of three disks. When a file is written to a RAID 5 volume, the file is split across all disks in the set except one. The final disk in the set has parity information written to it. Parity information is stored on each disk in the array. The generation of parity information makes writes to a RAID 5 array slower than writes to a RAID 0 volume. The benefit of RAID 5 is that the parity information stored on each disk allows the RAID 5 array to keep functioning when one of the disks that make up the volume fails. In terms of the RAID options that you can manage by using the Windows Server 2003 operating system, RAID 5 provides the best mix of redundancy and performance.

RAID 10

RAID 10 is a combination of RAID 1 and RAID 0. You implement RAID 10 at the hardware level rather than by using the Windows operating system. In RAID 10, you connect mirrored disk pairs together to form a RAID 0 array. For example, a RAID 10 array is built using six hard disk drives. Three mirrored sets are created and then data

is striped across these three mirrored sets. When a file is written, it is split equally between each of the mirrored sets. Because RAID 10 does not require parity calculations, it provides faster write speeds than RAID 5. For a RAID 10 set to fail, both disks of a mirrored pair would need to fail at the same time.

MORE INFO RAID levels

The following article provides more information about RAID: msdn.microsoft.com/library/default.asp?url=/library/en-us/optimsql/odp_tun_1_87jm.asp.

Failover Clustering

Failover clustering allows high-availability support for a SQL Server 2005 instance. The basic idea behind failover clustering is that by configuring more than one computer to host a single instance of SQL Server 2005, the database will continue to be available if one of the computers fails. Failover clusters consist of one or more servers with two or more shared disks. In the case of SQL Server 2005, the shared disks host the database data. Each server in a cluster is called a *node*; the combination of nodes and shared disks is termed a *resource group*; and the resource group, its network name, and its IP address are referred to as a *virtual server*.

To a client on the network, a SQL Server 2005 virtual server appears to be a single computer. If one node in the cluster fails, the application automatically fails over to another node in the cluster. From the perspective of the client on the network, nothing has changed and the database is still available to respond to queries.

You can have up to 25 virtual servers in a cluster, as each virtual server requires its own resource group, consuming a drive letter. Each virtual server is limited to a single instance of SQL Server. You use SQL Server Setup to create and configure virtual servers on failover clusters. You also use SQL Server Setup to add and remove nodes without affecting the other cluster nodes. When one of the nodes in the cluster fails, you can rejoin it to the cluster by running SQL Server Setup.

MORE INFO Failover clustering

For more information about failover clustering with SQL Server 2005, consult the following MSDN article: [msdn2.microsoft.com/en-us/library/ms189134\(SQL.90,d=ide\).aspx](http://msdn2.microsoft.com/en-us/library/ms189134(SQL.90,d=ide).aspx).

Quick Check

1. Which type of RAID offers good performance but no fault tolerance?
2. How many instances of SQL Server 2005 can you install on a single virtual server in a failover cluster?
3. A node in a SQL Server 2005 failover cluster crashes. After repairing it, you need to add it back to the failover cluster. What tool would you use to do this?

Quick Check Answer

1. RAID 0 provides good performance but no fault tolerance.
2. You can install only one instance of SQL Server 2005 on a virtual server in a failover cluster.
3. You would use SQL Server Setup in such a scenario.

Database Mirroring

Database mirroring is a method of increasing database availability. Mirroring works on the individual database level and can be implemented only if the full recovery model, covered in Lesson 3, “Recovering from Database Disaster,” is used. Database mirroring maintains two copies of a single database, each of which resides on a different server instance. The principal server responds to client requests, and the mirror server acts as a hot standby server. When the mirroring session is synchronized, database mirroring supports rapid failover with no loss of data or committed transactions.

When you implement database mirroring, all insert, update, and delete operations that occur on the principal database are redone on the mirror database as soon as possible. This task is accomplished by sending every active transaction log record to the mirror server, which applies these log records to the mirror database. Whereas replication works at the logical level, database mirroring works at the log record level.

Database mirroring has the following benefits:

- Increases data protection by providing complete or near complete data redundancy, depending on which operating mode you configure.
- Increases database availability in the event of disaster. In high-safety mode with automatic failover, you can bring the database online without data loss. In other operating modes, service can be forced to a standby copy with possible data loss.

Database Mirroring Operating Modes

Database mirroring uses three different operating modes:

- **High-safety mode** The mirror server synchronizes the mirror database with the principal database as fast as possible. After the databases are synchronized, a committed transaction is committed on both partners. This mode has a higher transaction latency than high-performance mode.
- **High-performance mode** The mirror server attempts to keep up with the log records sent by the principal server. Transactions commit on the principal without waiting for the mirror server to write the log to disk. This mode has lower transaction latency than high-safety mode, but there is a risk of data loss.
- **High-safety mode with automatic failover** This mode requires a third database instance termed a *witness*. The witness server verifies whether the principal server is functioning. The mirror server initiates automatic failover when the mirror and witness remain connected but neither can contact the principal server.

Role Switching

Role switching involves transferring the principal role to the instance functioning as a mirror. The mirror server takes over the principal role and brings its copy of the database online as the new principal. The former principal server assumes the mirror role. Role switching can take the following three forms:

- **Automatic failover** This option works only when a witness server is present and the high-safety mode is configured. (See the explanation of high-safety mode with automatic failover mentioned earlier.)
- **Manual failover** This option requires high-safety mode. The database must be synchronized.
- **Forced service** This option works with high-performance and high-safety modes. It can force service if the principal server has failed and the mirror is available. This role switch carries with it the possibility of data loss.

MORE INFO Database mirroring

For more information about database mirroring, consult the following article on MSDN: msdn2.microsoft.com/en-us/library/ms189852.aspx.

Mirroring with Failover Clustering

It is possible to use both mirroring and clustering. In this implementation, the principal server runs on the virtual server of one cluster, and the mirror server runs on the

virtual server of a different cluster. It is also possible to establish a mirror session where one partner resides on a cluster's virtual server and the other on an unclustered computer. The witness computer can run either on a third cluster or on an unclustered computer.

Log Shipping

Log shipping is the process whereby SQL Server automatically transfers transaction log backups from a primary database on a primary server instance to one or more secondary databases on separate secondary server instances. SQL Server applies these transaction log backups individually to each separate secondary database. A further option is to configure a third server to record the history and status of restore and backup operations. It is possible to configure this instance, termed the *monitor server*, to raise alert notifications in the event that these operations fail.

The log shipping process has three distinct phases:

1. Back up of the transaction log on the primary server instance
2. Transfer of the backed-up transaction log to one or more secondary server instances
3. Restoration of the transaction log on each secondary server instance

You cannot configure log shipping to automatically fail over from the primary to the secondary server. If the primary database fails, you must bring one of the secondary databases online manually.

Exam Tip Remember the precise requirements for automatic failover.

When using log shipping, you must configure the primary database to use the full or bulk-logged recovery model. Shifting to the simple recovery model causes log shipping to stop functioning. If there are multiple databases on a primary server that you want to configure to use log shipping, you must ship all to the same secondary server or group of servers.

BEST PRACTICES Using an extra computer to implement log shipping

If there are five separate SQL Server computers, each running an important database, one way of implementing log shipping is to purchase a sixth computer and to use it as the secondary server for each of the existing five computers. In the event of catastrophic hardware failure, the sixth computer can take over running the database until the original computer is restored.

Log Shipping Jobs

SQL Server Agent handles the following four types of log shipping jobs:

- **Backup job** Created for each primary database. This job runs every two minutes by default. It performs backup operations, logs history to the local and monitor servers (if configured), and deletes old backup files.
- **Copy job** Runs on each secondary server. This job copies backup files from the primary server and logs history on the secondary and monitor servers (if configured). The copy schedule should approximate the backup schedule.
- **Restore job** Runs on the secondary server, and restores copied backup files to secondary databases. This job logs history on the local and monitor servers. It deletes old files and history information. The frequency with which it runs depends on whether the secondary server will function as a warm standby or allows for catastrophic events to be detected, preventing them from being written to the secondary database.
- **Alert job** Created on a monitor server. This job raises alerts for primary and secondary databases when backup operations have not completed successfully.

The administrator determines how frequently log backups are taken, the rate at which they are copied to secondary servers, and the frequency at which they are applied to the secondary database. It is possible to copy and restore each transaction log backup on a secondary server moments after it is created. This approach reduces the amount of time required to bring a secondary server online. Another approach is to delay applying transaction log backups to the secondary database so that catastrophic actions on the primary database, such as a critical table being dropped, are not carried over to secondary servers.

Configuring Log Shipping

You can configure log shipping using SQL Server Management Studio (SSMS) or by executing a series of stored procedures. The configuration of log shipping involves the following general steps:

1. Identify which servers will function as primary, secondary, and optional monitor servers.
2. Create a file share for the transaction log backups on the primary server.
3. Select a backup schedule for the database on the primary server.

4. Create a file share on each secondary server where transaction log backups can be copied.
5. Configure the secondary databases.
6. Configure the optional monitor server.

The practice for this lesson contains a detailed walkthrough showing how to configure log shipping on the AdventureWorksDW database. Log shipping requires SQL Server 2005 Standard Edition, Workgroup Edition, or Enterprise Edition. All servers involved in log shipping must have the same case sensitivity settings, and the databases in the log shipping configuration must use the full or bulk recovery models.

Secondary Server Configuration

When configuring a secondary server, the installation path for the secondary server must be the same as the installation path for the primary server. It is also important to note that log shipping does not guarantee that no data will be lost. Any data that SQL Server has not backed up on the primary database and shipped to the secondary database is lost during failure. Backing up the service master key at the primary server is also important so that the key can be used on the secondary server in the event of failure.

MORE INFO Further details about log shipping

For more information about log shipping, consult the following article on MSDN: [msdn2.microsoft.com/en-us/library/ms187103\(d=ide\).aspx](http://msdn2.microsoft.com/en-us/library/ms187103(d=ide).aspx).

PRACTICE Configure Log Shipping

In this practice, you configure log shipping for the AdventureWorksDW database. You configure the Melbourne server as the primary server, and Glasgow as the secondary server.

1. On server Melbourne, using Microsoft Windows Explorer, create a folder named `c:\LogShip`.
2. Edit the properties of this folder to create a shared folder named LogShip.
3. Set the permissions of the LogShip folder so that the Everyone group has full control.

CAUTION Full control permissions

In the real world, granting the Everyone group full control to a share is highly inadvisable. In this demonstration, doing so simplifies the process.

4. Repeat this process on server Glasgow.
5. On server Melbourne, open SSMS.
6. Expand the Databases folder, and locate the AdventureWorksDW database.
7. Right-click the AdventureWorksDW database on the server Melbourne and choose Properties.
8. Under Select A Page, select the Options page.
9. Change the Recovery Model to Full and click OK. This closes the AdventureWorksDW Database Properties dialog box.
10. Right-click the AdventureWorksDW database on the server Melbourne and choose Properties to reopen the database Properties dialog box.
11. Under Select A Page, select the Transaction Log Shipping page.
12. Select the Enable This As A Primary Database In A Log Shipping Configuration check box.
13. Click Backup Settings. In the Network Path To Backup Folder text box, enter **\\Melbourne\LogShip**. In the Type A Local Path To The Folder field, enter **c:\logship** and then click OK.
14. Under Secondary Server Instances And Databases, click Add.
15. On the Secondary Database Settings page, click Connect.
16. In the Server Name text box, type **GLASGOW** and click Connect.

NOTE AdventureWorksDW database

Because the AdventureWorksDW database is already installed on server Glasgow, you do not need to re-create it.

17. On the Copy Files tab, in the Destination Folder For Copied Files text box, enter **\\GLASGOW\LogShip** as shown in Figure 4-1.
18. On the Restore Transaction Log tab, under Database State When Restoring Backup, verify that No Recovery Mode is selected.
19. Click OK.

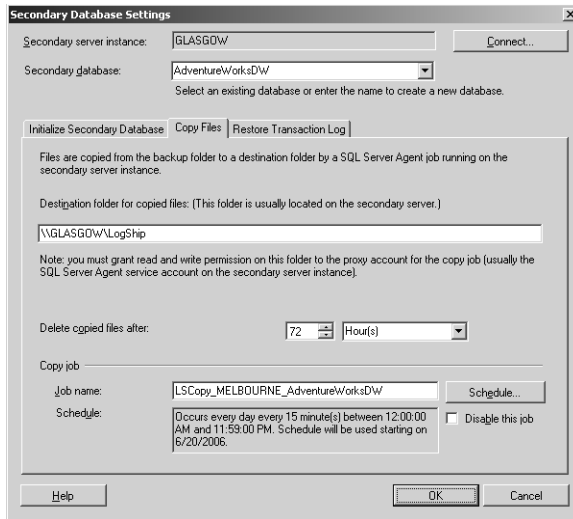


Figure 4-1 Secondary Database Settings dialog box.

20. Verify that the settings you configured in the Database Properties dialog box match those in Figure 4-2, and then click OK.

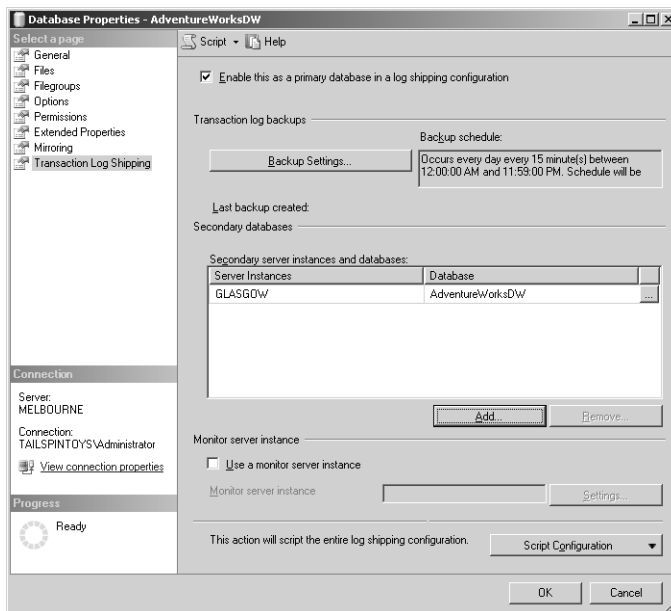


Figure 4-2 Transaction log shipping properties for the AdventureWorksDW database.

21. The Save Log Shipping Configuration dialog box should indicate that log shipping has been successfully configured. Click Close when the process completes.

Lesson Summary

- RAID 0 provides the best performance but also provides no redundancy. RAID 1 provides redundancy but offers no performance improvement over a normal disk drive. RAID 5 provides both redundancy and an improvement in performance over a normal disk drive.
- Failover clustering allows more than one computer to share an instance of SQL Server 2005. If one of the nodes in a cluster fails, another node ensures that the instance of SQL Server 2005 remains operational. The configuration of SQL Server 2005 virtual servers is accomplished through SQL Server Setup.
- Database mirroring maintains two copies of a single database, each of which resides on a different server instance. In database mirroring, each active transaction log record is sent to the mirror server, which applies these log records to the mirror database. Mirroring works at the log record level. Database mirroring can be implemented only if the full recovery model is in use.
- To allow automatic failover, database mirroring must use the high-safety mode and a monitor server. Database mirroring in high-performance mode has lower transaction latency, but there is a risk of data loss.
- Log shipping has transaction log backups from a primary database transfer automatically to a secondary server. These transaction log backups are then applied to the secondary server. Log shipping cannot be configured to automatically fail over from the primary server to the secondary server. Log shipping requires that the full or bulk-logged recover models be in use.

Lesson Review

You can use the following questions to test your knowledge of the information in Lesson 1, “Planning for Fault Tolerance.” The questions are also available on the companion CD if you prefer to review them in electronic form.

NOTE Answers

Answers to these questions and explanations of why each answer choice is correct or incorrect are located in the “Answers” section at the end of this book.

1. You are involved in discussions about purchasing a new server to host a SQL Server 2005 database for your company. You want to ensure that the operating system, SQL Server 2005 program files, and the transaction log are hosted on

fault-tolerant volumes. You also want to ensure that the database data files are hosted in such a way that they have the best possible mix of fault tolerance and performance. The operating system and program files can be located on the same volume as each other, but the transaction log should be located on a separate volume, as should the database files. If a hard disk drive can host only a single volume, what is the minimum number of disks required for this server?

- A. Three
 - B. Five
 - C. Six
 - D. Seven
2. Which of the following types of RAID should you NOT use as the basis for a volume to host database files if you want to ensure that the volume survives the failure of a hard disk drive?
- A. RAID 0
 - B. RAID 1
 - C. RAID 5
 - D. RAID 10
3. Failure of which of the following components brings down a two-node failover cluster running SQL Server 2005?
- A. Processor on the first node
 - B. Disk drive that hosts the operating system on the second node
 - C. Network interface cards on the first node
 - D. Shared disks
4. Which of the following need to be present for a mirrored database to be able to automatically fail over? (Choose all that apply.)
- A. Full recovery model
 - B. Monitor server
 - C. Failover cluster
 - D. High-performance mode

Lesson 2: Recovering from Failure

This lesson concentrates primarily on how to recover from a failure of SQL Server 2005 server. The most likely sort of SQL Server 2005 server failure that you will encounter is a corruption of the program files or the system databases. This lesson deals with the strategies that the database administrator can employ if either of these events occurs. The final part of this lesson provides information about the differences between database recovery models and various database backup methods.

After this lesson, you will be able to:

- Reinstall an instance.
- Determine whether to restore the system database.
- Understand the differences between recovery models.

Estimated lesson time: 30 minutes

Restoring the System Databases

If the system databases become corrupt, SQL Server 2005 is unlikely to start correctly. You have two basic options in this situation: restore the system databases from the regular backups that you have taken of them, or use the SQL Server 2005 installation media to rebuild the databases from scratch.

BEST PRACTICES Backing up system databases

Remember to back up your system databases as well as your user databases. Some administrators become so focused on protecting all the data in the user databases that they forget that important information is also stored within the system databases. You don't need to back up system databases as rigorously as user databases, but it is prudent to back them up on a reasonably regular basis. You should back up the master database after you create a database, change configuration values, or configure SQL logins. It is not possible to perform a differential backup on the master database.

If SQL Server cannot start because of a problem with the master database and backups of that database exist, start the server in single-user mode and execute the `RESTORE DATABASE` statement to restore the master full database backup. Remember, if changes were made to the master after the last backup was taken, you will lose those changes when you restore the master database. For example, logins that you created after you last backed up the master database will be lost. You will need to create them again by using either SQL Server Management Studio or any script that you

used to create the logins. Note that database users who were associated with lost logins will be orphaned and will be unable to access the database. After you restore the master database, the instance of SQL Server is halted.

If a backup of the system databases does not exist or has itself become corrupted, or single user mode cannot be reached, you can rebuild the system databases using the SQL Server 2005 installation media. To rebuild system databases for a default instance of SQL Server 2005 from the command prompt, perform the following steps:

1. Insert the SQL Server 2005 installation media into the drive. (These instructions assume that the DVD-ROM drive with the media is drive D.)
2. Open a command prompt, and issue the following command:

```
start /wait D:\setup.exe /qn INSTANCENAME=<MSSQLSERVER>  
REINSTALL=SQL_Engine REBUILDDATABASE=1 SAPWD=<NewStrongPassword>
```

Use the instance name MSSQLSERVER for the default instance. After you rebuild the system databases, you should be able to start SQL Server 2005 properly. From this point, perform a restore of the most recent full backup of the system databases. If you have attempted these strategies and SQL Server still will not start, you might need to repair all files and rebuild the registry.

To rebuild the registry of a damaged installation of SQL Server, you need to use the REINSTALL=ALL and REINSTALLMODE=OMUS parameters. Using these parameters rebuilds, repairs, and verifies a Microsoft SQL Server instance. Not only are system databases repaired, but program files and registry entries are repaired as well. It is important when you execute this operation that you ensure you use the same package file and options you specified during the original installation. If the original package file options are unknown, uninstall SQL Server completely and then perform a clean install rather than attempting to rebuild.

To repair all files and rebuild the registry at the same time, perform the following steps:

1. Insert the SQL Server 2005 installation media into the drive. (These instructions assume that the DVD-ROM drive with the media is drive D.)
2. Open a command prompt, and issue the following command:

```
start /wait D:\setup.exe /qb INSTANCENAME=MSSQLSERVER REINSTALL=ALL  
REBUILDDATABASE=1 REINSTALLMODE=omus SAPWD=<NewStrongPassword>
```

When you rebuild the system databases using the distribution media, all service packs and updates are lost. To expedite the rebuild process, you can use the /qn

switch in the command shown earlier to stop the display of setup dialogs and error messages. Rebuilding the master database places all system databases in their default locations. If in the original installation you moved these databases to alternate locations, you must manually move them again after the repair process is complete. As soon as possible, you should reapply these service packs and updates.

If the restored backup of a database is not current, you will need to re-create any missing entries manually. At this point, if you had previously configured the server instance as a replication Distributor, you need to restore the distribution database as well.

Recovery Models

Database recovery models determine the granularity to which it is possible to restore a database in the event of a failure. There are three possible recovery models, and each has its benefits and drawbacks. Each recovery model is detailed in the next few pages.

NOTE Default recovery models

The master, msdb, and tempdb databases use the simple recovery model by default. The model database uses the full recovery model by default.

Simple Recovery Model

The simple recovery model has the least administrative overhead because it does not require you to back up the transaction log. When you implement the simple recovery model on a database, it is not possible to back up the transaction log and SQL Server automatically truncates the transaction log (that is, it drops inactive parts of the log to free up space) after each backup completes.

If the database is damaged, you can recover data only to the point at which it was backed up. For example, if the database is backed up only once a day at 3 A.M., and the database fails at 11 P.M., you will lose 20 hours of updates. When using the simple recovery model, you must strike a balance between the impact that backing up has on performance and the size of the window of time in which it will not be possible for you to recover data.

The simple recovery model is most appropriate for development databases or databases containing read-only data, such as data warehouses. If the loss of recent changes to the database is unacceptable, you should not use the simple recovery model.

Full Recovery Model

The full recovery model is the best possible option to prevent data loss. The full recovery model requires database and transaction log backups. The full recovery model also provides protection against media failure. To ensure that you do not lose transactions, you should host the transaction log on a fault-tolerant volume such as RAID 1 or RAID 5.

SQL Server 2005 allows you to back up the transaction log during a normal data or differential backup. The Enterprise Edition of SQL Server 2005 allows you to restore a database without the necessity of taking it offline. You can perform an online restore using SQL Server Enterprise Edition only if you have configured the database to use the full or bulk-logged recovery models.

Bulk-Logged Recovery Model

The bulk recovery model is a subset of the full recovery model. Under this model, SQL Server minimally logs bulk operations such as index creation or bulk imports. This arrangement improves performance and log space consumption; otherwise, SQL Server would need to enter each individual transaction in its entirety into the log. The drawback of the minimal logging of such operations is that only minimal records of the individual transactions that occur in the bulk operation exist. This means that it is impossible for you to perform a point-in-time restore. When a log backup contains bulk-logged operations, you can recover the database only to the end of the log backup. If something goes wrong in the middle of the bulk operation, it is impossible for you to run the log forward to a point just prior to the failure. The bulk-logged recovery model is in some respects an all-or-nothing approach.

To back up a log that contains bulk-logged operations, you need access to the data files that contain the bulk-logged transactions. This added data can make the log backup large.

Before implementing the bulk recovery model, you should be aware of the following restrictions:

- If you make a filegroup containing bulk-logged changes read-only before performing a log backup, all subsequent log backups of that filegroup will contain the extents changed by the bulk operation as long as the filegroup remains read-only. A way of avoiding this limitation is to implement the full recovery model prior to making the filegroup read-only and backing up the log. After the backup is finished, you should then make the filegroup read-only.

- If a log backup contains bulk-logged changes, you cannot perform a point-in-time recovery.
- If bulk operations are performed while the bulk-logged recovery model is implemented, all files must be online or defunct during log backup.
- Online restore sequences work only if all log backups were taken prior to the damage and the bulk changes were backed up prior to starting the online restore sequence.

Selecting a Recovery Model for a Database

Configuring a database to use a specific recovery model is relatively simple. To configure a database's recovery model, perform the following steps:

1. Open SQL Server Management Studio.
2. Expand the Databases folder.
3. Right-click the database for which you want to configure the recovery model, and choose Properties.
4. In the Select A Page pane, select the Options page.
5. Use the Recovery Model drop-down list to select the chosen recovery model, as shown in Figure 4-3. Click OK to save your changes.

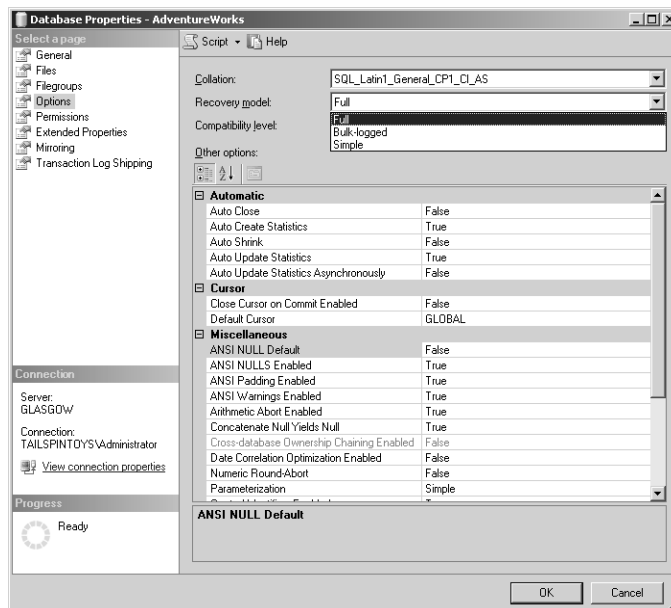


Figure 4-3 Configuring file recovery options.

Quick Check

1. What factors can make a bulk recovery log backup large?
2. Which recovery model should you use if it is always necessary to restore to a particular named transaction?
3. In what situation should you NOT use the REINSTALL=ALL and REINSTALLMODE=OMUS repair options?

Quick Check Answer

1. Bulk recovery logs require access to the data files that contain the bulk-logged transactions.
2. The full recovery model should be used in this situation.
3. You should NOT use these repair options if you are unaware of which package options were used for the SQL Server installation.

Files and Filegroups

Understanding how files and filegroups work in SQL Server 2005 is an important part of understanding how to recover from a failure of SQL Server 2005. SQL Server databases are not stored as large binary blobs of information on a disk, but rather are stored as individual files and filegroups.

Files

Database data and log information are never collected in the same file. Individual files are never shared between databases. A SQL Server 2005 database uses three types of files:

- The primary data file is the database starting point. Each database has a single primary data file. Primary data files use the .mdf extension.
- Secondary data files host all data that is not located in the primary data file. Not all databases have secondary data files, and you must explicitly create them. Using secondary data files can improve performance. You often place secondary files on different volumes from the one that stores the primary data file. Secondary data files use the .ndf extension.
- Log files host information used in the recovery of databases. Each database must have at least one log file. Log files use the .ldf extension.

Both the database's primary file and the master database keep records of the location of each file that constitutes an individual database. SQL Server 2005 does not enforce the file name extensions mentioned in the preceding list, though these extensions are helpful for identifying different kinds of files.

SQL Server 2005 files are identified by both a logical and physical file name. The differences between these names are as follows:

- **logical_file_name** A nickname you assign to the file. You use this nickname to refer to the file in Transact-SQL statements. The logical file name must comply with the rules for SQL Server identifiers. It must be unique among logical file names in the database.
- **os_file_name** Name of the file and its location in the file system.

For example, when you install it using the default settings, the AdventureWorks primary database file has the two names listed in Table 4-1.

Table 4-1 Logical and Physical File Names

logical_file_name	AdventureWorks_Data
os_file_name	C:\Program Files\Microsoft SQL Server\MSSQL.1\MSSQL\Data\AdventureWorks_Data.mdf

The state of a file is independent of the state of the database. Files can have the six states described in Table 4-2.

Table 4-2 SQL Server 2005 File States

State	Definition
ONLINE	The file is available for all operations.
OFFLINE	The file is not available for access. Files are set offline when they are corrupted but can be restored. You can set an offline file to online only by restoring it from backup.
RESTORING	The file is being restored.
RECOVERY PENDING	Recovery of the file has been postponed. Files enter this state because of a piecemeal restore process in which the file is not restored and recovered.

Table 4-2 SQL Server 2005 File States

State	Definition
SUSPECT	File recovery failed during online restore. If the file is located in the primary filegroup, SQL Server marks the database as suspect. The file remains in this state until you make it available via restore and recovery, or DBCC CHECKDB with REPAIR_ALLOW_DATA_LOSS.
DEFUNCT	The file was dropped when offline. When an offline filegroup is removed, all files in that filegroup become defunct.

Filegroups

Database filegroups allow you to group database objects and files together. SQL Server 2005 uses the following two types of filegroups:

- **Primary filegroup** Contains the primary data file and all other files that are not assigned to other filegroups. System tables and their pages are allocated to the primary filegroup.
- **User-defined filegroups** Contains tables, indexes, and large object data that have been associated with a specific filegroup. You can place data in partitioned tables and indexes in separate filegroups to improve performance.

Log files are not part of any filegroup and are treated as objects distinctly different from database files. One filegroup in each database is set as the default filegroup. The default filegroup stores all tables or indexes that you create without specifying a filegroup. If you do not configure a default filegroup, the primary filegroup automatically takes this role.

Filegroups are online only if all files within the filegroup are online. Files in the primary filegroup are online if the database is online. If a file in the primary filegroup goes offline, the database goes offline. Any attempt to access an offline filegroup with a Transact-SQL statement results in an error.

You can mark all filegroups, except the primary filegroup, read-only. A read-only filegroup cannot be modified. You should store tables that contain fixed data, such as those in a data warehouse, in read-only filegroups. Read-only filegroups also support NTFS compression. Note, though, that compression reduces performance, and you should consider compression only in cases where the data is not accessed often. SQL Server system databases such as master, model, msdb, resource, and tempdb do not support compression.

Backup Types

SQL Server 2005 allows four categories of backup: data backup, differential backup, transaction log backup, and copy backup.

NOTE Backups

Although the objectives for Exam 70-444 do not actually mention the word *backup*, a description is provided in this book to make understanding the mechanics of possible restore operations easier.

Data Backups Data backups include the entire image of one or more data files or filegroups. There are three kinds of data backups:

- **Database backup** A full backup of the entire database. This includes a part of the transaction log, which allows the full database backup to be recovered. These backups are complete and represent the entire database at the time the backup completed. You can estimate the size of a full database backup by using the *sp_spaceused* system stored procedure.
- **Partial backup** Partial backups differ from database backups in that they do not contain all the filegroups. Partial backups must contain all data in the primary filegroups, every read-write filegroup, and nominated read-only files. A partial backup of a read-only database contains only the primary filegroup.
- **File backup** File backups are often used to increase recovery speed by allowing only the restoration of damaged files rather than the restoration of the entire database. This is useful when the database contains several files located on different volumes and one of those volumes fails. In such a situation, you need to restore only the file on the failed volume.

Differential Backups Differential backups are taken in relation to a specific differential base. Differential bases include a full database backup, a partial database backup, or a file backup. Occasionally, differential backups are taken in relation to a set of bases (multibased differential). A differential backup covers the same set of files as the base backup, but it backs up only the extents that have changed since the base was created by the original data backup. The three types of differential backups are as follows:

- **Differential database backup** Backs up all files in the database containing extents modified since the most recent database backup
- **Differential partial backup** Backs up data extents from a previous partial backup that were modified since that partial backup was taken
- **Differential file backup** Backs up files containing data extents changed since the most recent full database backup of each file

Transaction Log Backups You can perform transaction log backups only during a full database backup. There are three types of transaction log backups:

- **Pure log backup** Contains only transaction log records for an interval. A pure log backup does not contain bulk changes performed under the bulk-logged recovery model.
- **Bulk log backup** Includes log records and data pages changed by bulk operations. Point-in-time recovery on bulk-logged backups is disallowed.
- **Tail-log backup** A backup taken just before restoring a database to capture records that have not been backed up. Tail-log backups have the following properties:
 1. They are the last backups of interest in a recovery plan.
 2. Restoring a database without performing a tail-log backup results in an error unless the restore statement contains either the `WITH REPLACE` or `WITH STOPAT` clauses.
 3. They can be attempted if the database is damaged or not online. They succeed when the log files are undamaged and the transaction log does not contain any bulk-logged changes.
 4. They can be created independently of regular log backups by using the `COPY_ONLY` option. The transaction log is not truncated when using the `COPY_ONLY` option.
 5. They can contain incomplete metadata if the database is damaged. This can occur if the log backup is taken with `CONTINUE_AFTER_ERROR`, because backup occurs independently of the state of the database.

Truncating the Transaction Log Truncating a transaction log is the process whereby you free up space in the log by deleting the inactive parts of the log. Active parts of the log are unaffected by truncation. If all parts of the transaction log are active, truncation is delayed. When using the bulk-logged or full recovery model, inactive parts of the transaction log are truncated after each automatic checkpoint. Automatic checkpoint intervals are based on the amount of log space used and the time elapsed since the last checkpoint. If few modifications are made to the database, the time interval between checkpoints can be long. If the database is constantly being modified, automatic checkpoints occur more frequently.

After an initial data backup has occurred, SQL Server does not truncate inactive parts of the transaction log until they have been backed up. This happens regardless of any automatic checkpoints that might occur. The only way to stop the transaction log

from filling is to ensure that you regularly back it up. Backing up the transaction log automatically truncates it, freeing up space.

To determine how full the transaction log is, use the DBCC SQLPERF (LOGSPACE) command as shown in Figure 4-4.

	Database Name	Log Size (MB)	Log Space Used (%)	Status
1	master	0.4921875	64.28571	0
2	tempdb	0.7421875	53.88158	0
3	model	0.4921875	57.14286	0
4	msdb	0.7421875	65.78947	0
5	ReportServer	0.7421875	56.38158	0
6	ReportServerTempDB	0.7421875	60.85526	0
7	AdventureWorks	1.992188	21.37255	0
8	AdventureWorksDw	1.992188	39.21569	0
9	distribution	1.992188	36.20098	0
10	University	0.9921875	55.56102	0

Figure 4-4 Checking the amount of transaction log space.

Copy Backups Copy backups are a feature new to SQL Server 2005, and they do not affect the overall backup and restore procedures. For example, you might have a backup scheme where you take a full database backup every third day and differential backups every six hours. From the perspective of the next differential backup, taking a copy backup of the full database does not count as a database backup, and SQL Server backs up only extents that have changed since the original full database backup.

You often use copy-only backups when preparing an online restore because a copy-only backup does not affect the sequence of regular transaction logs. You should note that you cannot directly perform a copy backup using the SQL Server Management Studio interface. You can accomplish the creation and restoration of copy-only backups only by using Transact-SQL statements. COPY_ONLY backups are recorded in the backupset table in the is_copy_only column.

Snapshot Backups

Snapshot backups provide an almost instantaneous copy of the data that they back up. Snapshot backups require the installation of extra software and hardware from independent software vendors on the computer running SQL Server 2005. The

benefit of snapshot backups is that they minimize the use of SQL Server resources in accomplishing the backup. Snapshot backups work either by splitting a mirrored disk set or by creating a copy of a disk block at the time of writing.

NOTE Database snapshots and snapshot backups

Although they are similarly named, do not confuse snapshot backups with database snapshots. The name *snapshot backups* is evocative of the speed of the backup. A *database snapshot* is an image of the database taken at a specific point in time.

The benefits of snapshot backups are as follows:

- Creates an almost instantaneous backup with minimal impact on the server
- Restores backup almost instantaneously
- Backs up to tape on another host without affecting the performance of the production system
- Generates a copy of the production database almost instantly for the purposes of reporting or testing

You cannot perform an online restore from a snapshot backup. Restoring a snapshot backup takes the database offline. Piecemeal restores can incorporate snapshot backups, but all restore sequences will be offline restores. Snapshot backups are tracked in the msdb database and are identified by the entry `backupset.is_snapshot = 1`.

PRACTICE Use the Full Recovery Model and Back Up a Database

In this practice, you configure the AdventureWorks database to use the full recovery model. You then take a full database backup. After you take the full database backup, you perform a differential backup of the database.

1. If necessary, open SSMS and connect to the local instance.
2. Expand the Databases folder.
3. Right-click the AdventureWorks database and choose Properties. This opens the Database Properties dialog box.
4. In the Select A Page pane, select the Options page.
5. Ensure that the recovery model is set to Full and then click OK.
6. Right-click the AdventureWorks database, choose Tasks, and then choose Back Up. This opens the Back Up Database dialog box, as shown in Figure 4-5.

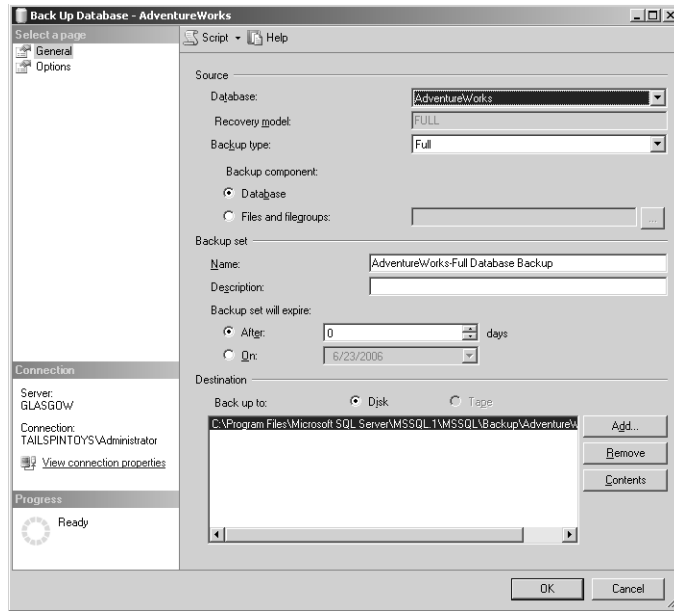


Figure 4-5 The Back Up Database dialog box.

7. Ensure that the Backup Type drop-down list shows Full as the selected option, and ensure that the Backup Set section shows the Name selected as AdventureWorks-Full Database Backup. Click OK.
- The backup process takes several moments to complete.
8. Click OK to close the Successful Database Backup message box.
 9. Right-click the AdventureWorks database, choose Tasks, and then choose Back Up. This again opens the Back Up Database dialog box.
 10. Set the Backup Type to Differential, and then click OK to start the backup.
 11. Click OK in the successful database backup message box.
 12. Right-click the AdventureWorks database, choose Tasks, and then choose Back Up. This again opens the Back Up Database dialog box.
 13. Change the Backup Type to Transaction Log.
 14. In the Select A Page pane, select the Options page.
 15. Ensure that the Truncate The Transaction Log option button is selected.
 16. Click OK to start the transaction log backup.
 17. Click OK in the Successful Database Backup message box.

Lesson Summary

- You can restore the system databases in single-user mode and use the SQL Server 2005 installation media.
- If SQL Server 2005 still does not start after you have rebuilt the system databases, you might need to rebuild the registry and repair the instance. You can do this by using the REINSTALL=ALL and REINSTALLMODE=OMUS parameters from the start command of the SQL Server media.
- The simple recovery model requires the least administrative effort because it does not require you to back up the transaction log. When you implement this model, SQL Server automatically truncates the transaction log after each backup completes. The simple recovery model does not allow point-in-time or named transaction restore. Restore is possible only to the point where the database was backed up.
- The full recovery model provides the best option for recovering data. It supports point-in-time, named transaction, and online restore. The transaction log is truncated only after it is backed up.
- The bulk logged recovery model is similar to the full recovery model except that it will only minimally log bulk operations. The disadvantage of this model is that it cannot provide point-in-time or named transaction restore unless no bulk operations have occurred since the log was last backed up.
- A database backup is a full backup of the entire database. This includes a part of the transaction log, which allows the full database backup to be recovered. These backups are complete and represent the entire database at the time the backup completed.
- Partial backups differ from database backups in that they do not contain all the filegroups. Partial backups must contain all data in the primary filegroups, every read-write filegroup, and nominated read-only files.
- File backups are often used to increase recovery speed by allowing the restoration of only damaged files rather than restoring the entire database.
- Truncating a transaction log is the process whereby space is freed up in the log by deleting the parts of the log that are inactive.

Lesson Review

You can use the following questions to test your knowledge of the information in Lesson 2, “Recovering from Failure.” The questions are also available on the companion CD if you prefer to review them in electronic form.

NOTE Answers

Answers to these questions and explanations of why each answer choice is correct or incorrect are located in the "Answers" section at the end of the book.

1. You are unsure of which options were used when SQL Server 2005 was installed. An error on the disk has caused many program files to become corrupt. Which course of action should you pursue to get SQL Server 2005 working again?
 - A. Restore the system databases in single-user mode.
 - B. Use the installation media to rebuild the system databases.
 - C. Use the installation media with the REINSTALL=ALL and REINSTALLMODE=OMUS switches.
 - D. Uninstall SQL Server 2005, and perform a clean reinstall.
2. Which recovery model always allows the database administrator to recover to a particular named transaction?
 - A. Simple
 - B. Bulk-logged
 - C. Full
3. Under which circumstances can a file in the primary filegroup be offline?
 - A. The database is online.
 - B. The database is offline.
 - C. The primary filegroup is marked read-only.
 - D. The primary filegroup is compressed.
4. What sort of backup should you always try to take prior to performing a database restoration when a database uses the full recovery model and is online?
 - A. Tail-log backup
 - B. Bulk log backup
 - C. Full database backup
 - D. Differential backup

Lesson 3: Recovering from Database Disaster

This lesson concentrates on how to recover from a SQL Server 2005 database disaster. A database disaster can be anything from an inadvertently dropped table full of important data to a database that has had its constituent files hopelessly corrupted. Just as there are many types of backups, there are many methods of restoration, each of which is appropriate for a particular set of circumstances. The simplest form of restoration is to restore to the last backup. The more complex methods involve restoring backups and applying transaction logs. The technique that a database administrator employs is tied directly to the desired target recovery point. The more specific the target point, the more complicated the restoration process.

After this lesson, you will be able to:

- Perform an online restore.
- Restore to a point in time.
- Restore to a named transaction.
- Revert to a database snapshot.
- Restore to include the most recent transaction log.
- Configure lost logins.
- Recover lost data.

Estimated lesson time: 30 minutes

Restoration and Rolling Forward

Restoring involves extracting data from a backup and then applying logged transactions to that data to bring it forward to the target recovery point. The target recovery point is the state in which you want the data when you complete the restoration process. *Rolling forward* is the term used for the process of applying logged changes to data in a database to bring it forward to the target recovery point. Data and differential backups contain enough transaction log records to allow you to either roll forward with the active transactions or to roll back uncommitted transactions so that the database is in a consistent state.

A roll forward set includes one or more full, partial, or file data backups. Unless specified otherwise, all files in the backup being restored will be included in the roll forward set. You can use the RESTORE statement to limit the roll forward set to specific filegroups, files, or pages. If the data backup contains log records, data will be rolled forward using these log records.

Restore Sequences

You execute restore scenarios using one or more restore operations, termed a *restore sequence*. Restore sequences are either a set of Transact-SQL RESTORE statements or restore plans generated by SQL Server Management Studio. If a problem occurs with the restore sequence—for example, a database administrator overshoots his intended recovery point—you must restart the restore sequence from the very beginning. In such a situation, you would need to restore the backups again and then start rolling the log forward again.

Restore Phases

Under the full recovery model, you can specify the recovery point as a specific point in time, a specific transaction, or a log sequence number. The bulk-logged recovery model allows point-in-time restore, but only if no bulk operations have been performed since the previous backup.

Restoration often involves the following three distinct phases:

- **Data copy phase** Data, log, and index pages are copied from the backup media to the database files. This might involve restoring a full database backup and then performing appropriate differential backups. After these items have been restored, you need to reset the contents of the affected database, files, or pages to the time that they were captured by the backups.
- **Redo phase** This phase applies logged transactions to the data copied to the database during the data copy phase, rolling that data forward to the recovery point. During the redo phase, it is common for a database to have uncommitted transactions and to be in an unusable state.
 1. If you are restoring the primary file, the recovery point dictates the state of the entire database. For example, if you are recovering a database to a point of time just prior to an important table being dropped, you must return the entire database to that point in time.
 2. If you are not restoring the primary file, restored data is rolled forward to a point consistent with the database.
 3. If data was read-only when it was backed up, roll forward is unnecessary.
 4. The final part of the redo phase is applying the tail-log backup.
- **Undo phase** In this final phase, SQL Server identifies uncommitted transactions and rolls them back to bring the database to a consistent state. The undo phase completes a restore sequence, and you cannot restore subsequent backups as a

part of that particular sequence. If a needed file or page is offline and an uncommitted transaction cannot be rolled back, the transaction becomes deferred. Deferred transactions occur only when the specific data needed by the roll back process is offline during startup. When the undo phase completes, the database comes back online and is accessible to users.

Prior to performing a restoration, you need to perform a tail-log backup. You can accomplish this task through SQL Server Management Studio as discussed in Lesson 2, “Recovering from Failure,” or by using the BACKUP command with the NO_TRUNCATE option.

MORE INFO Fast recovery

Fast recovery is a feature unique to SQL Server 2005 Enterprise Edition, and it is available during crash recovery or database mirroring failover. Transactions that were yet to be committed when the failure occurred regain the locks held prior to the failure. As these transactions are rolled back, the locks stop the possibility of user changes.

Restore Types

There are several types of restore: complete database restore, file restore, page restore, and piecemeal restore.

Complete Database Restore

In the full and bulk-logged recovery models, a complete database restore involves restoring a full database backup, any relevant differential backup, and all sequential log backups taken after the last differential (or if there are no differential, full database) backup including the tail-log backup. The database is offline for the duration of the restore. Prior to bringing the database online, you must recover the data to a consistent point.

The general steps for performing a complete database restore are as follows:

1. Perform a tail-log backup.
2. Restore the most recent full database without recovering the database (RESTORE WITH NORECOVERY).
3. Restore the most recent differential backup if it exists (RESTORE WITH NORECOVERY).
4. Start with the first transaction log backup created after the backup that was just restored. Restore the logs in sequence, including the tail log, with NORECOVERY.

5. Recover the database (RESTORE DATABASE <db_name> WITH RECOVERY).

MORE INFO Performing database restores

For more information about performing a complete database restore, consult the following MSDN article: msdn2.microsoft.com/en-us/library/ms187092.aspx.

File Restore

File restores enable restoration of one or more damaged files without requiring the restoration of the entire database. All editions of SQL Server 2005 support offline file restore. The Enterprise Edition of SQL Server 2005 allows the restoration and recovery of files to an offline secondary filegroup while a database is online. After you have recovered all files in the offline filegroup, the filegroup comes back online.

The general steps in performing a file restore are as follows:

1. Create a tail-log backup of the active transaction log.
2. Restore each damaged file from the most recent backup of that file (WITH NORECOVERY).
3. Restore the most recent differential file backup for each restored file if it exists (WITH NORECOVERY).
4. Restore the transaction log backups taken after the most recent restored backup file in order, from oldest to most recent (WITH NORECOVERY).
5. Restore the tail-log backup (WITH RECOVERY).

MORE INFO Performing file restores

For more information about performing a file restore, consult the following MSDN article: msdn2.microsoft.com/en-us/library/ms190710.aspx.

Page Restore

Page restores are useful in repairing isolated damaged pages. Online page restore uses the improved page-level error reporting available in SQL Server 2005 Enterprise Edition. Online page restore is not supported in other editions of SQL Server 2005. You use page restores when a small number of pages have been damaged. Restoring several individual pages is often faster than doing a file restore. Page restores have the advantage of reducing the amount of data that is offline during a restoration. If you need to restore many pages, performing a file restore is a more appropriate action.

Page restore is not supported by the simple recovery model. You can restore only database pages. Page 0 of all data files (the file boot page), page 1:9 (the database boot page), and any full text catalogs cannot be restored using a page restore. If you cannot successfully restore a page, a file or full database restore is necessary.

The general steps in performing a page restore are as follows:

1. Obtain the identification of the torn pages that need restoration. Lesson 4, “Salvaging Data from a Damaged Database,” covers procedures that you can use to accomplish this.
2. Locate the full, file, or filegroup backup that contains the page.
3. In the RESTORE DATABASE statement, use the PAGE clause to list all page IDs that require restoration, as shown in the following example:

```
RESTORE DATABASE <database> PAGE='1:42, 1:84, 1:126'  
FROM <file_backup >  
WITH NORECOVERY;
```

4. Apply any available differential backups required for the pages to be restored (with NORECOVERY).
5. Create a new log backup of the database that includes the final log serial number of the restored pages.
6. Restore the new log backup (with RECOVERY).

MORE INFO Performing page restores

For more information about performing page restores, consult the following MSDN article: msdn2.microsoft.com/en-us/library/ms175168.aspx.

Piecemeal Restore

During a piecemeal restore, you recover the database on a filegroup-by-filegroup basis. The primary filegroup is restored first. If a file is undamaged and is consistent with the database, it does not need to be restored and is recovered immediately. In an online piecemeal restore, the database comes online after you restore the primary filegroup.

The first restore must use a full database backup that contains the primary filegroup, specify the PARTIAL option, and have enough transaction log records to allow the database to reach a state of consistency. Using the partial option on a later restore statement will begin a new piecemeal restore scenario.

MORE INFO Performing piecemeal restores

For more information about performing piecemeal restores, consult the following MSDN article: msdn2.microsoft.com/en-us/library/ms177425.aspx.

Restoring a Database to a Point in Time

You usually use a point-in-time restore to restore the database to a time just before a catastrophic failure has occurred. You can use the point-in-time restore method to restore to the following points:

- A particular time within a transaction log
- A named mark inserted into the transaction log
- A specific log sequence number

A point-in-time restore is similar to other restores except that prior to restoring the backup that contains the target point in time, you must specify that point of time as the recovery point for the restore sequence.

For a point in time, use the following:

```
RESTORE { DATABASE | LOG } <database name> FROM <backup_device> WITH STOPAT = <time>
```

To restore only modifications before a specific point in time, include the `NORECOVERY` option.

When restoring to a specific marked transaction, the two available options are `STOPATMARK` and `STOPBEFOREMARK`:

- `STOPATMARK` includes the marked transaction in the roll forward.
- `STOPBEFOREMARK` runs right up to, but does not include, the marked transaction.

Here is an example of how to use the `STOPATMARK` option:

```
RESTORE { DATABASE | LOG } <database name> FROM <backup_device>  
WITH STOPATMARK = '<mark_name>'
```

Restoring to a particular log sequence number uses the `STOPATMARK` and `STOPBEFOREMARK` syntax, except rather than using '`<mark_name>`', you use the '`!sn:<lsn_number>`' clause. `STOPBEFOREMARK` rolls forward to the log sequence number, but it excludes that particular log record from the roll forward. In the following

example, the transaction log for the AdventureWorks database is applied up until, but not including, log serial number 42:

```
RESTORE LOG AdventureWorks FROM DISK = 'c:\adventureworks_tlog.bak'  
WITH STOPBEFOREMARK = 'lsn:42'
```

Restoring Databases When SQL Server Is Offline

You can recover and restore a database using SQL Writer if the SQL Server is offline. If a full-text catalog is associated with the database, you must stop the Microsoft Full-Text Engine for SQL (MSFTESQL) service prior to attempting restoration.

Safety Features

SQL Server 2005 includes several safety features that ensure that you cannot overwrite a database with a restore operation from another database. For example, suppose that Tailspin Toys has two databases installed on a single SQL Server 2005 instance—one for sales and another one for human resources. The built-in safety features of SQL Server 2005 ensure that it is impossible to overwrite the sales database by restoring files from the human resources database. These safety features activate and cause the restore operation to fail if the following conditions are in effect:

- The database in the restore operation is not the same as the database recorded in the backup set. The feature is sophisticated enough to distinguish a different database even if the databases have the same name, as long as each database was created differently.
- The restore operation needs to create automatic files, but the files that it needs to create already exist on the server.

Quick Check

1. While attempting a restore to a point prior to a transaction that dropped a table, a database administrator accidentally overshoots when applying the transaction log. What steps does the DBA need to take to get back to the point prior to the table being dropped?
2. What are the names of the three distinct restore phases?
3. If SQL Server is offline and an attempt is being made to restore a database that includes a full text catalog, which service must be stopped prior to attempting the restoration?

Quick Check Answer

1. The database administrator must begin the restore sequence from scratch.
2. The three distinct restore phases are data copy, redo, and undo.
3. The Microsoft Full-Text Engine for SQL Service must be stopped.

Database Snapshots

A database snapshot is a read-only static view of a database as it exists when the snapshot is created. A database snapshot does not include transactions uncommitted at the time the snapshot was taken. Database snapshots must be located on the same server instance as the database. You can store multiple snapshots of a database, and snapshots persist until dropped. Database snapshots are supported by all recovery models. Snapshots are not direct copies of a database. Database snapshots work at the data-page level. Before a data page of the source database is modified for the first time since the database snapshot was taken, that page is copied from the source database to the snapshot. The snapshot stores this original data page, preserving the data records as they existed at snapshot creation. Any subsequent updates to that data page are not copied. This means that a database snapshot grows each time that a data page is modified from the state it was in at the time of the database snapshot's creation. Because subsequent updates are not copied to the snapshot, a snapshot can never grow larger than the source database at the time of snapshot creation.

The copies of the original pages are stored in sparse files. At the time you take a database snapshot, the sparse file is empty. As data pages are updated, the sparse file grows. If you do not properly plan the storage of snapshots, the database snapshot might run out of space. If this happens, the snapshot is marked as suspect and must be dropped.

The following list details the benefits of snapshots:

- Snapshots can be queried. This allows report generation to be based on the data that exists at the time of snapshot creation.
- In the event of user error on a source database, you can revert the database to the state it was in when the snapshot was created. Data loss is then confined to updates made to the database since the snapshot's creation. For high-level protection, you can create a series of database snapshots. For example, you could implement 12 rolling snapshots over a 24-hour period. Each time a new snapshot is created, the oldest snapshot can be deleted.

- You can manually reconstruct lost data from information in the snapshot. You can also bulk copy data from the snapshot into the database and then manually merge it.
- Data can be safeguarded prior to bulk updates by taking a snapshot prior to the bulk update operation. Reverting is quicker than restoring from backup, though it is not possible to roll forward from this point.

Snapshots are reliant on the source database and therefore cannot be used as a substitute for taking backups. For example, you can use a snapshot to recover information accidentally deleted from the database, but database snapshots are useless in recovering data lost if a volume that hosts a database fails.

Database snapshots have the following limitations:

- Snapshots are a feature that is available only in SQL Server 2005 Enterprise Edition.
- While a snapshot exists, you cannot drop, detach, or restore a database.
- While a snapshot exists, you cannot drop files from the source database or from any snapshots.
- While a snapshot exists, the source database must be online (unless the database is a mirror database within a database mirror session).
- While a snapshot exists, you cannot configure the source database as a scalable shared database.
- Database snapshots work on an entire database.
- You cannot take snapshots of the model, master, or tempdb databases.
- Snapshots are read-only.
- You cannot back up, restore, attach, or detach snapshots.
- You cannot create snapshots on FAT32 file systems.
- Database snapshots do not support full-text indexing.
- Snapshots inherit the security constraints of the source database. These permissions cannot be changed because snapshot databases are read-only.
- Snapshots reflect the state of filegroups at the time of creation. It is impossible to change the status of a filegroup that is offline in a snapshot to online.
- When using log shipping, you can create database snapshots only on the primary database. If you switch roles, you must drop all database snapshots before you set up the primary database as a secondary.
- Read-only or compressed filegroups cannot be reverted.

Each database snapshot requires a unique database name. Database snapshots can be created by any user with the permissions to create a database. To create a database snapshot, use the following Transact-SQL code:

```
CREATE DATABASE DatabaseSnapshotName ON
( NAME = DatabaseName, FILENAME =
'C:\Program Files\Microsoft
SQL Server\MSSQL.1\MSSQL\Data\DatabaseSnapshotName_Data.ss' )
AS SNAPSHOT OF DatabaseName;
GO
```

Where *DatabaseName* is the name of the database that the snapshot is being taken of, *DatabaseSnapshotName* is the database snapshot name and *DatabaseSnapshotName_Data.ss* is the name and location where the snapshot will be stored.

The practice at the end of the lesson describes how to make a snapshot of the AdventureWorks database and then how to restore the database back to that snapshot.

To view the database snapshots that exist on a server, perform the following steps:

1. Open SQL Server Management Studio.
2. Expand the Databases folder.
3. Expand the Database Snapshots folder as shown in Figure 4-6.

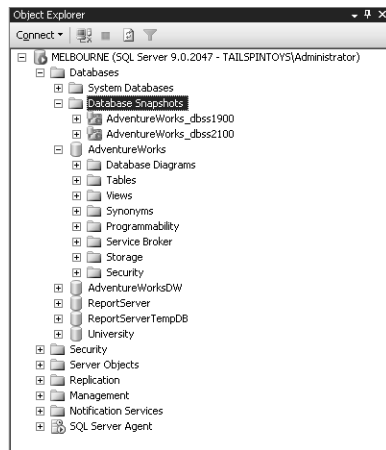


Figure 4-6 Expanded Database Snapshots folder showing two database snapshots of the AdventureWorks database.

Any user with the RESTORE DATABASE permission on the source database can revert the database to its snapshot state. Reverting overwrites all updates made to the database since the creation of the database snapshot. The revert operation overwrites

the log file and rebuilds the log. This means that it is impossible to roll the reverted database forward to the point of user error. Reverting to a snapshot drops all full-text catalogs. Reverting is unsupported under the following conditions:

- Files are offline that were online when the snapshot was made.
- More than one snapshot of the database currently exists.

During a reversion, the snapshot and source databases are unavailable. If an error occurs during a reversion, the revert operation attempts to complete when the database restarts. Although you cannot restore the original log to roll forward, you should back it up prior to reverting a snapshot as a guide to reconstructing lost data. Because reverting breaks the log backup chain, you must take a full database or file backup before log backups can resume.

Troubleshooting Orphaned Users

An orphaned user is a database user for which the SQL Server login is undefined. Database users can become orphaned if the following conditions exist:

- The corresponding SQL Server login is dropped.
- A database is restored or attached to a different instance.
- The database user is mapped to a SID that is not present on the server instance.

Exam Tip On the exam, you are most likely to encounter orphaned users in restore scenarios, where user accounts were created after the backup was taken.

You can locate orphaned users by using the following Transact-SQL statement:

```
sp_change_users_login @Action='Report';
```

Executing this statement outputs the list of users and SIDs in the current database that are not linked to any SQL Server login. Note that *sp_change_users_login* does not work with logins created from users' Windows accounts.

You can relink server login accounts with a database user by using the following Transact-SQL statement:

```
sp_change_users_login @Action='update_one',
@UserNamePattern='<DatabaseUser>'
@LoginName='<ServerLogin>';
```

If a user who connects to the database using her Windows login becomes unable to connect to the database after you restore the MASTER database, you will need to create a new login for that user with the CREATE LOGIN Transact-SQL statement and map it to her Windows login.

PRACTICE Create and Revert to a Snapshot

In the first part of the practice, you make a snapshot of the AdventureWorks database. You will call this snapshot Adventureworks_dbss01. In the second part of the practice, you restore the database to the state it was in when you took the snapshot.

1. If necessary, open SQL Server Management Studio, connect to the appropriate instance, and expand the Databases folder.
2. Right-click the AdventureWorks database and choose New Query.
3. In the new Query window, enter the following:

```
CREATE DATABASE AdventureWorks_dbss01 ON
( NAME = AdventureWorks_Data, FILENAME =
'C:\Program Files\Microsoft SQL
Server\MSSQL.1\MSSQL\Data\AdventureWorks_data_01.ss' )
AS SNAPSHOT OF AdventureWorks;
GO
```

4. Right-click the Database Snapshots folder and choose Refresh.
5. Expand the Database Snapshots folder, and verify that the Adventureworks_dbss01 snapshot is present.

NOTE Reverting to a snapshot

Reverting to a database snapshot requires that any database snapshots other than the one that you are reverting to be dropped.

6. Close all query windows without saving any queries.
7. Right-click the Databases folder and choose New Query.
8. In the New Query window, enter the following code:

```
RESTORE DATABASE AdventureWorks FROM DATABASE_SNAPSHOT =
'AdventureWorks_dbss01'
```

9. Verify that the query executed successfully.

Lesson Summary

- Restoring involves extracting data from a backup and then applying logged transactions to that data to bring it forward to the target recovery point.
- If a problem occurs with the restore sequence—for example, a database administrator overshoots the intended recovery point—it is necessary to restart the restore sequence from the very beginning.
- Restoration involves three distinct phases: the data copy phase, redo phase, and undo phase. During the data copy phase, data, log, and index pages are copied from the backup media to the database. During the redo phase, logged transactions are applied to the database. During the undo phase, uncommitted transactions are identified and rolled back to bring the database to a consistent state.
- File restores enable restoration of one or more damaged files without requiring the restoration of the entire database. Page restores are used when a small number of pages have been damaged. Restoring several individual pages is often faster than performing a file restore.
- A database snapshot is a read-only static view of a database as it existed when the snapshot was created. You can store multiple snapshots of a database, and snapshots persist until dropped. You can query a database snapshot. You can revert the database to the state it was in when the snapshot was taken. You can bulk copy data from the snapshot into the database and then manually merge it. Snapshots cannot be backed up or restored, nor attached or detached.
- An orphaned user is a database user for which the SQL Server login is undefined.
- The *sp_change_users_login* stored procedure can be used to re-link orphaned users who use SQL Server 2005 authentication, but not users who use Windows for authentication.

Lesson Review

You can use the following questions to test your knowledge of the information in Lesson 3, “Recovering from Database Disaster.” The questions are also available on the companion CD if you prefer to review them in electronic form.

NOTE Answers

Answers to these questions and explanations of why each answer choice is correct or incorrect are located in the “Answers” section at the end of the book.

1. Contoso Ltd. has a Windows Server 2003 computer with a SQL Server 2005 instance installed. The Mineralogical database runs on this instance. The computer is configured so that the transaction log and the Mineralogical database files are stored on separate hard disk drives. The Mineralogical database is configured to use the full recovery model, with full backups taken weekly and differential backups taken every six hours. The transaction log is backed up on a two-hour basis. Thirty minutes before the next transaction log backup, the volume hosting the Mineralogical database fails. The database administrator locates a replacement hard disk drive and installs it. She also locates the necessary backup media containing the full backup, the most recent differential backup, and the transaction log backups that have occurred since the most recent differential backup. What step should the Contoso database administrator take prior to performing a restoration to ensure that the database is restored in the most complete manner possible?
 - A. Run DBCC CHECKDB.
 - B. Truncate the transaction log.
 - C. Perform a tail-log backup.
 - D. Perform a full database backup of the Mineralogical database.
2. You are the database administrator for Contoso. You have configured the Mineralogical database so that a snapshot of the database is created every two hours. Snapshots older than 24 hours are automatically dropped. You get a telephone call from a panicked manager who informs you that one of the developers at Contoso has accidentally dropped three tables containing critical mineralogical analysis data in the last half an hour. A database snapshot was taken 45 minutes ago. You attempt to revert the database to the snapshot taken 45 minutes ago but are unable to. Which of the following steps must you take to do this?
 - A. Drop all snapshots except the one taken 45 minutes ago.
 - B. Restart SQL Server 2005 in single-user mode.
 - C. Take the Mineralogical database offline.
 - D. Run DBCC CHECKDB.
3. Several system databases have become corrupt. It is possible to restore all except the MASTER database from the most recent backup. The MASTER database is restored from a tape that is more than a month old. After the restoration, three users report that they are unable to log on to the Mineralogical database using

their Windows accounts. Which of the following strategies should you use to allow these users to log in to the database?

- A. Restore the Mineralogical database.
- B. Use the *sp_change_users_login* stored procedure.
- C. Use the CREATE LOGIN Transact-SQL statement.
- D. Execute the DBCC CHECKDB stored procedure.

Lesson 4: Salvaging Data from a Damaged Database

Even though most of a backup tape is perfectly fine, a few crinkled inches can mean that all the data on the tape is unavailable. Gigabytes of data might be lost just because the tape drive didn't eject the backup tape correctly. SQL Server 2005 provides tools that enable a DBA to restore the data on the parts of the tape that are still fine. This lesson also focuses on the SQL Server 2005 tools that you can use to repair corrupted data stored within the database.

After this lesson, you will be able to:

- Restore data from bad tapes.
- Repair corrupt data.

Estimated lesson time: 30 minutes

Restoring Data from Bad Tapes

You encounter restore errors if the backup media is in a damaged state. There are many ways in which media can be damaged, from dust getting on the tape to a bad eject from the tape drive. Restore errors are either reported by Windows or detected by checksums. When an error is encountered, the restore terminates immediately. This can mean that it is impossible to restore a database because the data is held on a slightly damaged tape that keeps generating an error when you make an attempt to extract data from it. When attempting to restore data from backup media that is damaged, there are three courses of action:

- Attempting to repair the problem causing the error and restart the restoration
- Allowing the restoration to continue despite errors, and then attempting to repair the database after the restore completes
- Attempting an alternate recovery plan that avoids the damaged backup

Occasionally, you can resolve media errors by doing something as simple as cleaning the tape drive. Most tape drives require regular maintenance, and cleaning the drive might restore full functionality. It might also be necessary to replace the tape drive or to attempt to restore the data elsewhere. You should remember that it is just as important to follow the vendor's guidelines for maintaining backup hardware as it is to take backups in the first place.

If the media errors aren't resolved by running a head cleaning product over the tape drive, SQL Server 2005 introduces a new RESTORE option, `CONTINUE_AFTER_ERROR`, that you can use to extract good data from a bad tape. This option allows a restore operation to continue despite errors, enabling restoration of all possible data from the damaged media.

The roll forward process continues, and you can apply subsequent transaction log backups. Any errors that the roll forward process encounters that prevent the process from reaching the target point in time are indicated in the log. Pages that fail verification when `CONTINUE_AFTER_ERROR` is used are written to disk and logged in the `suspect_pages` table within the `msdb` database and the SQL Server error log.

MORE INFO Responding to restore errors

For more information about responding to restore errors caused by damaged backups, consult the following article on MSDN: [msdn2.microsoft.com/en-us/library/ms190952\(SQL.90,d=ide\).aspx](http://msdn2.microsoft.com/en-us/library/ms190952(SQL.90,d=ide).aspx).

Quick Check

1. What is the name of the new RESTORE option that you can use to extract intact data from a partially damaged tape?
2. Where is information about pages that fail verification logged?

Quick Check Answer

1. `CONTINUE_AFTER_ERROR` is the name of the RESTORE option used to extract intact data from a partially damaged tape.
2. Information about pages that fail verification is logged in the `suspect_pages` table in the `msdb` database and the SQL Server error log.

Using DBCC CHECKDB to Repair Data

If you are unable to restore data from backup, you can use the `DBCC CHECKDB` command as a repair option of last resort. Executing `DBCC CHECKDB` performs an examination of the allocation, structural, and logical integrity of all objects in the database. When you run `DBCC CHECKDB`, the following actions occur:

- `DBCC CHECKALLOC` is run on the database.
- `DBCC CHECKTABLE` is run on every table and view in the database.

- DBCC CHECKCATALOG is run on the database.
- Service Broker is validated.
- The contents of all indexed views in the database are validated.

When you run DBCC CHECKDB, it is not necessary to run DBCC CHECKALLOC, DBCC CHECKTABLE, or DBCC CHECKCATALOG. In some situations, you can repair the database using DBCC CHECKDB with the WITH TABLOCK option. To determine the minimum level of repair needed, run DBCC CHECKDB without a repair option against the database. This action will recommend which repair option to use. Even though the database has been damaged, back up the database prior to using the repair options in the event they make things worse.

To use the repair options, the database must be in single-user mode. The available repair options are as follows:

- **REPAIR_ALLOW_DATA_LOSS** Tries to repair all reported errors, but can result in some data loss
- **REPAIR_FAST** Maintains syntax for backward compatibility, but no repair actions are performed
- **REPAIR_REBUILD** Performs all repairs possible without risk of data loss

If a damaged table has one or more constraints, it is recommended that you run DBCC CHECKCONSTRAINTS when the repair operation is complete. In some cases, the damage might be so severe that there is not enough information to repair the database. To gain limited access to the remaining data, it will be necessary to place the database into emergency mode. You can accomplish this by using the EMERGENCY option of the ALTER DATABASE command. EMERGENCY mode allows members of the sysadmin fixed server role read-only access. This allows them to diagnose problems and retrieve any available data.

Rebuilding Indexes

In some situations, it might be necessary for you to rebuild an index. It might be necessary because the index has become corrupt, or because it is necessary to remove fragmentation and reorder the index in contiguous pages. Heavily fragmented indexes can significantly degrade query performance. There are several ways of rebuilding indexes. The first method is to use SQL Server Management Studio; the second and third methods involve issuing Transact-SQL statements.

To rebuild an index using the SQL Server Management Studio, perform the following steps:

1. Expand the Databases folder.
2. Expand the database that contains the table with the specified index.
3. Expand the Tables folder to locate the table in which the index resides.
4. Expand the table in which the index resides and then expand the Indexes folder.
5. Right-click the index you want to rebuild and choose Rebuild. This opens the Rebuild Indexes dialog box. Alternatively, to rebuild all indexes on a table, right-click the Indexes folder within the table and select Rebuild All.
6. Click OK to start the Index rebuild.

There are two options when rebuilding an index using Transact-SQL statements:

- ALTER INDEX with the REBUILD clause
- CREATE INDEX with the DROP_EXISTING clause

Both methods achieve the same result, dropping the existing index and then creating a new one.

MORE INFO Differences between rebuilding options

For more information about the differences between ALTER INDEX with REBUILD and CREATE INDEX with DROP EXISTING, consult the following MSDN article: msdn2.microsoft.com/en-us/library/ms189858.aspx.

Managing Suspect Pages

The `suspect_pages` table in the `msdb` database holds information about suspect pages. A suspect page is a database page containing an 824 error. The `suspect_pages` table records the page ID as well as the event type. Possible event types are listed in Table 4-3.

Table 4-3 `suspect_error` Table Event Types

Error Description	Event_type Value
824 errors other than bad checksum or torn page	1
Bad checksum	2
Torn page	3

Table 4-3 suspect_error Table Event Types

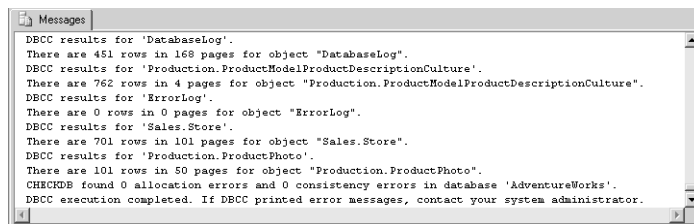
Error Description	Event_type Value
Restored	4
Repaired (DBCC Repaired)	5
Deallocated by DBCC	7

Running DBCC CHECKDB REPAIR_ALLOW_DATA_LOSS in single-user mode updates the suspect_pages table indicating whether SQL Server deallocated or repaired the suspect page. The suspect_pages table is limited in size. After the table fills, SQL Server does not log new errors. You need to manually delete rows so that SQL Server can write new entries to this table. In some situations, an administrator might want to update a record—for example, to mark a suspect page as intact.

PRACTICE Check a Database for Errors

In this practice, you use the DBCC CHECKDB utility to check the AdventureWorks database for any possible errors.

1. If necessary, open SQL Server Management Studio and connect to the appropriate instance.
2. Expand the Databases folder.
3. Right-click the AdventureWorks database and choose New Query.
4. In the Query window, type **DBCC CHECKDB** and click the Execute button.
5. When the command finishes executing, scroll to the bottom of the messages window and verify that it has the same message shown in Figure 4-7.



```
DBCC results for 'DatabaseLog'.
There are 451 rows in 168 pages for object "DatabaseLog".
DBCC results for 'Production.ProductModelProductDescriptionCulture'.
There are 762 rows in 4 pages for object "Production.ProductModelProductDescriptionCulture".
DBCC results for 'ErrorLog'.
There are 0 rows in 0 pages for object "ErrorLog".
DBCC results for 'Sales.Store'.
There are 701 rows in 101 pages for object "Sales.Store".
DBCC results for 'Production.ProductPhoto'.
There are 101 rows in 50 pages for object "Production.ProductPhoto".
CHECKDB found 0 allocation errors and 0 consistency errors in database 'AdventureWorks'.
DBCC execution completed. If DBCC printed error messages, contact your system administrator.
```

Figure 4-7 Output result of the DBCC CHECKDB command on the AdventureWorks database.

Lesson Summary

- Restore errors are either reported by Windows or detected by checksums.
- The `CONTINUE_AFTER_ERROR` restore option means that an attempt will be made to continue extracting data from the backup media even when an error is encountered.
- Pages that fail verification when `CONTINUE_AFTER_ERROR` is used are written to disk and logged in the `suspect_pages` table and error log.
- `DBCC CHECKDB` performs an examination of the allocation, structural, and logical integrity of all objects in the database.
- To use the `DBCC CHECKDB` repair options, the database must be in single-user mode.
- If the database damage is so severe that you cannot restore information, you might need to place the database into emergency mode.
- You can rebuild an index using SQL Server manager, or you can use the `ALTER INDEX` with `REBUILD` or `CREATE INDEX` with `DROP EXISTING` Transact-SQL statements.
- The `suspect_pages` table in the `msdb` database holds information about database pages containing an 824 error.
- Running `DBCC CHECKDB REPAIR_ALLOW_DATA_LOSS` in single-user mode will update the `suspect_pages` table, indicating whether the suspect page was deallocated or repaired.

Lesson Review

You can use the following questions to test your knowledge of the information in Lesson 4, “Salvaging Data from a Damaged Database.” The questions are also available on the companion CD if you prefer to review them in electronic form.

NOTE Answers

Answers to these questions and explanations of why each answer choice is correct or incorrect are located in the “Answers” section at the end of the book.

1. A database on a Windows Server 2003 computer running SQL Server 2005 is not online. Under which conditions will a tail-log backup be unsuccessful? (Choose all that apply.)
 - A. The database is configured to use the full recovery model.
 - B. The log files are damaged.
 - C. The log files contain bulk-logged changes.
 - D. The transaction logs are stored on a RAID 0 volume.
2. You have attempted to restore data from a tape but have encountered errors. You run a head cleaner through the tape drive, but this does not resolve the problem. Which of the following RESTORE options should you use to extract the maximum amount of good data from the tape?
 - A. STOP_ON_ERROR
 - B. CONTINUE_AFTER_ERROR
 - C. CHECKSUM
 - D. ERROR_BROKER_CONVERSATIONS
3. DBCC CHECKDB includes which of the following DBCC commands? (Choose all that apply.)
 - A. DBCC CHECKALLOC
 - B. DBCC CHECKTABLE
 - C. DBCC CHECKCATALOG
 - D. DBCC SHOWCONTIG
4. You have restored data from the backups, but there are still errors in one of the user databases. Which of the following methods should you try in an attempt to repair the maximum amount of data?
 - A. Start SQL Server in single-user mode. Execute the DBCC CHECKDB command with the REPAIR_ALLOW_DATA_LOSS option.
 - B. Start SQL Server in single-user mode. Execute the DBCC CHECKDB command with the REPAIR_REBUILD option.
 - C. Start SQL Server normally. Execute the DBCC CHECKDB command with the REPAIR_ALLOW_DATA_LOSS option.
 - D. Start SQL Server normally. Execute the DBCC CHECKDB command with the REPAIR_REBUILD option.

Chapter Summary

- Failover clustering allows more than one computer to share an instance of SQL Server 2005. If one of the nodes in the cluster fails, another node ensures that the instance of SQL Server 2005 remains operational.
- SQL Server Setup can be used to add and remove nodes from the cluster without adversely affecting the operation of the other nodes in the cluster.
- Database mirroring maintains two copies of a single database, each of which resides on a different server instance. To allow automatic failover, database mirroring must use the high-safety mode and a monitor server.
- Log shipping has transaction log backups from a primary database transfer automatically to a secondary server. These transaction log backups are then applied to the secondary server.
- Restoring involves extracting data from a backup and then applying logged transactions to that data to bring it forward to the target recovery point.
- A database snapshot is a read-only static view of a database as it existed when the snapshot was created.
- DBCC CHECKDB performs an examination of the allocation, structural, and logical integrity of all objects in the database. To use the DBCC CHECKDB repair options, the database must be in single-user mode.

Key Terms

Do you know what these key terms mean? You can check your answers by looking up the terms in the glossary at the end of the book.

- database mirroring
- differential base
- extent
- failover clustering
- full recovery model
- hot standby server
- log sequence number
- mirror server
- multibased differential

- node
- page
- principal server
- recovery point
- resource group
- restore sequence
- rolling forward
- sparse files
- truncation
- virtual server
- witness server

Case Scenarios

In the following case scenarios, you will apply what you've learned about planning for fault tolerance and disaster recovery. You can find answers to these questions in the “Answers” section at the end of this book.

Case Scenario 1: Ensuring Fault Tolerance

You have been asked to help design a high availability solution for Contoso, Ltd., which wants to install a SQL Server 2005 Enterprise Edition database solution to keep track of its human resource assets and product inventory. Hardware will be purchased according to your recommendations, though there is an insitutional distaste for server clusters and hardware RAID-based solutions.

1. The managers at Contoso want to ensure that the best fault tolerance and performance are achieved. The operating system, program files, transaction log, and database files should all be located on different volumes. No two volumes should be hosted on a single disk drive. Design a solution to meet these requirements that uses a minimum number of hard disk drives.
2. The managers at Contoso want to ensure that if a table is accidentally dropped, a restore operation can always be performed that will bring the database up to the state it was in a minute prior to the deletion. Which recovery model should be implemented on the database?

3. The managers at Contoso want to ensure that automatic failover occurs, but you do not want to implement failover clusters. What other high-availability technology could you implement?

Case Scenario 2: Backup and Recovery

You have been brought in as a consultant to Tailspin Toys, which has just moved from another vendor's database solution to SQL Server 2005. You are being retained by the company for several weeks because the in-house database administrator is not fully aware of the capabilities of SQL Server 2005.

1. How would you describe the differences between database mirroring and log shipping to the Tailspin Toys administrator?
2. One of the databases on the server will have a SQL Server Integration Services (SSIS) task that will import 20,000 records into the database each day. Which recovery model should you recommend the Tailspin Toys administrator implement on this database?
3. When attempting to restore data from backup tape, the Tailspin Toys administrator discovers several kinks in the backup tape. With the previous vendor's database solution, she would normally throw the tape in the trash. What advice can you give her about recovering as much data as possible from this tape?

Suggested Practices

To successfully master the exam objectives presented in this chapter, complete the following practice tasks:

Plan for Fault Tolerance

- **Practice 1: Configure transaction log shipping.** Create a basic database on server Melbourne, and then configure transaction log shipping so that the database is created on server Glasgow from a backup taken automatically on server Melbourne.
- **Practice 2: Configure recovery models.** Configure the AdventureWorks database to use the full recovery model.

Salvage Good Data from a Damaged Database by Using Restoration Techniques

- **Practice 1: Perform a check for torn pages.** Use the appropriate stored procedure to perform a check for torn pages in the AdventureWorks database.

Recover from a Database Disaster

- **Practice 1: Perform a full database restore.** Create a full database backup of the AdventureWorksDW database. Restore the database back to the server from the backup.

Recover from a Failure of SQL Server 2005

- **Practice 1: Recover using a snapshot.** Create a snapshot of the AdventureWorks database. Drop the Production.Document table. Revert to the snapshot you created prior to dropping the table.

Take a Practice Test

The practice tests on this book's companion CD offer many options. For example, you can test yourself on just one exam objective, or you can test yourself on all the 70-444 certification exam content. You can set up the test so that it closely simulates the experience of taking a certification exam, or you can set it up in study mode so that you can look at the correct answers and explanations after you answer each question.

MORE INFO Practice tests

For details about all the practice test options available, see "How to Use the Practice Tests" in this book's Introduction.

Chapter 5

Performance Monitoring

Monitoring server performance is an essential part of database administration, and Microsoft SQL Server has several tools that you can use to perform this task. The Microsoft Windows Server 2003 Performance tools, System Monitor, and Performance Logs and Alerts provide counters for SQL Server 2005 that allow you to track server resources and activities. SQL Server Profiler enables you to trace and analyze SQL Server events. Other tools and resources include stored procedures, distributed management views (DMVs), and the SQL Server log.

You need to plan your SQL Server performance monitoring strategy carefully. We saw in Chapter 1, “Troubleshooting Database and Server Performance,” that you can monitor performance counters and use SQL traces and DMVs to troubleshoot server, instance, and database performance. However, your monitoring strategy should accomplish more than troubleshooting. It should enable you to track trends, discover what resources are coming under more pressure through time, and solve problems before they occur. By tracking user activity and baselining performance at typical, busy, and quiet periods, you should be able to identify normal operation and distinguish between an unusual activity peak and persistent and permanent pressure on resources.

Another common reason for wanting to monitor SQL Server is to improve server performance. To achieve optimal performance, you need to minimize the time it takes for users to see the results of queries and maximize the total number of queries that the server can handle simultaneously. You do this by resolving hardware issues that might be degrading performance. If you can prove that an investment of a few hundred dollars in disk hardware can save thousands of dollars by improving throughput, you are doing your job as a database administrator (DBA), although you might not, technically, be troubleshooting.

Unfortunately, you often need to make tradeoffs when it comes to resource usage. For example, as the number of users accessing SQL Server grows, you might not be able to reduce the network traffic load, but you might be able to improve server performance by optimizing queries or indexing. Chapter 2, “Analyzing Queries,”

discusses how to detect badly formed queries or incorrectly indexed tables that use an excessive amount of resources or take a long time to complete, and how to improve the performance of these queries. However, you need to do more than troubleshoot bad queries. You need to monitor all queries and look at ways of optimizing performance.

Exam objectives in this chapter:

- Define and implement monitoring standards for a physical server.
 - Establish the thresholds for performance.
 - Establish the baselines for performance.
 - Define which types of information to monitor on the physical server.
 - Define traces.
 - Set alerts.
 - Set notifications.
- Choose the appropriate information to monitor.

Lessons in this chapter:

- Lesson 1: Defining and Implementing Monitoring Standards for a Physical Server 252
- Lesson 2: Choosing the Appropriate Information to Monitor 277

Before You Begin

To complete the lessons in this chapter, you must have completed the following tasks:

- Configured a Windows Server 2003 R2 computer with SQL Server 2005 Enterprise Edition SP1 as detailed in the Appendix.
- Installed an updated copy of the AdventureWorks sample database as detailed in the Appendix.

No additional configuration is required for this chapter.

Real World*Ian Mclean*

Monitoring should enable you to better plan your budgets and maintenance. Rather than finding yourself urgently needing additional memory to solve a problem, you should be able to present a business case to convince (often skeptical) management to include plans for a memory upgrade in the next budget. For this reason (among others), you should store your results in a database or in a comma-delimited format (.cdf) file, which enables you to produce graphs that illustrate increasing usage trends. You need to be able to argue the business case for investment in upgrades to hardware and software even when the technical case is self evident—because it might be self evident only to you. You need to answer the senior executive (and there is always one) who asks, “Why should we spend money on the database system? It never gives us any problems.”

Lesson 1: Defining and Implementing Monitoring Standards for a Physical Server

This lesson discusses setting *baselines* and establishing *threshold* values for alerts. It defines the types of information that you need to monitor on the physical server to capture events, check server health, and ensure maximum availability of both the server and the SQL Server 2005 services. The lesson investigates the distinction between physical and logical servers in a Microsoft Cluster Server (MSCS) environment and the differences in their monitoring processes.

The lesson covers the use of SQL traces, which you can generate by using SQL Server Profiler or stored procedures, and how you can use the Transact-SQL DBCC TRACEON statement to enable trace flags. It covers the types of alerts you can set and how you can trigger *notifications*.

NOTE Service packs

The service pack level at the time of this writing is Service Pack 1 (SP1). Unless otherwise indicated, all the information in the chapter applies to both SQL Server 2005 and SQL Server 2005 SP1.

After this lesson, you will be able to:

- Define monitoring standards and configure monitoring for the physical server.
- Capture performance logs that you can use as baselines for server performance.
- Set alerts to warn you when specified counters exceed or fall below threshold values.
- Distinguish between physical and logical servers in an MSCS environment.
- Generate traces to assist you in monitoring and assessing the performance of SQL queries and applications.

Estimated lesson time: 60 minutes

Establishing Performance Thresholds

The SQL Server 2005 services operate in a dynamic environment. The stored data, the type of access that users require, and the way that users connect all change over time. SQL Server automatically manages system-level resources such as memory and disk space, so the need for extensive system-level manual tuning is minimized. However, you need to identify performance trends to determine whether changes are necessary, or when they are likely to become necessary.

You monitor databases to assess how a server is performing. Monitoring involves taking periodic snapshots of current performance to isolate processes that could cause problems, and gathering information continuously over time to track performance trends. By monitoring the response times for frequently used queries, you can determine whether changes to the query or indexes on the tables are required; by monitoring users trying to connect to an instance of SQL Server, you can determine whether security is set up adequately and test applications; by monitoring SQL queries as they are executed, you can determine whether they are written correctly and producing the expected results.

Typically, however, the results you collect during the monitoring process are not absolute. Exceptions can occur—if you suddenly find you are running out of disk space, you need to take immediate action—but mostly the monitoring results are compared with the same readings taken at the same periods of activity at an earlier date or a series of dates. In other words, results are compared against baselines. If a counter value is increasing or decreasing over time, you need to detect this trend and decide whether it is a cause for concern. A baseline helps you find out whether the results are changing. To determine whether you need to take any action, you need to establish performance thresholds. Exceeding such a threshold is not necessarily the trigger for immediate (and expensive) hardware upgrades, but a trend that approaches a threshold needs to be factored into future planning. That having been said, some threshold conditions do require a prompt response, and you need to consider the appropriate alert configurations.

MORE INFO Monitoring and tuning for performance

For more information, search for “Monitoring and Tuning for Performance” in Books Online or access [msdn2.microsoft.com/en-us/library/ms189081\(d=ide\).aspx](https://msdn2.microsoft.com/en-us/library/ms189081(d=ide).aspx).

Determining Monitoring Strategy

To determine your monitoring strategy and set monitoring thresholds, you need to clearly define your goals, specify your baseline conditions, and select the appropriate components to monitor and the tools with which to monitor them. You need to select *metrics* for those components. Too much data is as bad as too little, and you want to measure only what is relevant. Your baseline conditions and the performance changes that occur over time enable you to establish thresholds, diagnose specific performance problems, and identify components or processes to optimize.

A monitoring strategy also enables you to audit user activity, test a server under different loads, design maintenance schedules, and test backup and restore plans. Finally, as thresholds are reached and hardware changes and upgrades become necessary, you can determine the most cost-effective modifications to your hardware configuration.

The Windows operating system and SQL Server 2005 provide a set of tools to monitor servers in the SQL Server environment. The Windows Performance tools, System Monitor, and Performance Logs and Alerts let you collect data about activities such as memory, disk, and processor usage and view this data in real-time or as collected in a log file. You can configure alerts to trigger on threshold counter values and perform predefined tasks—for example, sending a message and starting a transaction log backup. The Task Manager tool lets you take an instant snapshot of random access memory (RAM), the central processor unit (CPU), and network usage and the processes that are using these resources.

SQL 2005 Server provides SQL Trace. You can generate traces by using SQL Server Profiler or Transact-SQL stored procedures, and you can set trace flags by using Database Console Command (DBCC) statements—for example DBCC TRACEON. SQL Server Management Studio (SSMS) provides Activity Monitor and the Query Editor graphical execution plan. The Showplan event classes in Profiler also let you monitor and analyze query plans, and DMVs such as *dm_exec_query_stats* let you analyze query statistics.

You need to identify the components that you want to monitor, and you need to select the performance counters or Profiler event classes to include in your monitoring strategy and the counter instances or specific data about events captured in Profiler that are of interest to you. In Profiler, you can also specify filters so that you focus on the events pertinent to the monitoring scenario.

When you have decided what to measure and what tools to use, you can gather data during periods of peak activity, during quiet periods, or throughout a full working day. Based on these baselines—and on experience, stress testing, or results viewed over a period of time—you can analyze the data, decide on your action thresholds, and decide whether you need to configure an alert if a threshold is exceeded. Unfortunately, no precise formula exists for this activity. Every SQL Server installation has its own particular features, and every DBA has his or her own methods and opinions.

Establishing Performance Baselines

To determine whether your SQL Server system is performing optimally, you need to take performance measurements at regular intervals over time to establish a server performance baseline and to track trends. A professional DBA performs this task even

when no problems occur. You compare each new set of measurements with those taken previously and with the original readings. If you make major hardware changes, introduce new client applications, or make major changes to your database structure, you need to capture a new set of baselines.

IMPORTANT Schedule your data inspection.

Automating the data collection process is relatively straightforward. Finding time to inspect and analyze the data is often more difficult. In a similar way to security auditing, inspecting performance data is an invisible task. Only you know you are doing it—or whether you are doing it regularly. It is all too easy to get sidetracked into firefighting and ignore data inspection, and nobody will notice until your database system collapses. Schedule the job, reserve time for it, and do it on a regular basis.

Your baselines should include peak and off-peak measurements. To generate baseline measurements, you need to monitor the following items:

- **Hardware resources** For example: Memory: Pages/sec, Memory: Available Bytes, Physical Disk: % Disk Time, and Processor: % Processor Time
- **Network usage** For example: Network Interface: Bytes Total/sec, and Network Interface: Output Queue Length
- **Operating system activity** For example: System: Processor Queue Length and various instances of Process: %Processor Time
- **Server availability** For example: SQLServer:Buffer Manager: Buffer Cache Hit Ratio, SQLServer: General Statistics: User Connections, SQLServer: Databases: Transactions/sec for a specific application database, and SQLServer: Access Methods: Full Scans/Sec
- **Query performance** For example: SQLServer: Locks: Average Wait Time (ms)
- **Response and completion times** For example: Production-query or batch-command response times, and database backup and restore completion times

You use the same counters that you use to establish your baselines to measure current server performance. Counter values significantly above or below your baseline require further investigation, as do readings that are increasing or decreasing through time. They might indicate areas in need of tuning or reconfiguration, or hardware resources that are gradually coming under stress, and you should plan to upgrade or replace.

Real World

Ian Mclean

A baseline can mean different things to different people. I once worked with a colleague who was tasked with implementing a new client/server application that involved the installation of a new database on a SQL Server 2005 server. She told me she was establishing a baseline. Some time later, when users were complaining that the application was running slowly, I asked her for her baseline statistics and subsequent performance logs. I got a blank look. She meant a permissions baseline, which she generated by querying the `sys.database_permissions` and `sys.database_principals` tables in the new database. So we knew that the permissions were set correctly. We had no idea why performance was deteriorating.

Deciding What to Monitor

Two schools of thought exist about what you should monitor. Some experienced DBAs are selective about the counters and instances they recommend. Too much information can be a disadvantage, and logging can use resources and generate large files. Others, equally experienced, take the view that you should select the objects of interest and capture all counters and instances of these objects. If you save the information to a database, you can then use database queries or write your own application to access whatever information you need. In the latter case, you need to ensure you have enough disk space available and archive your data as needed. If the overhead is unacceptable, you can reduce your time interval instead of reducing the number of counters tracked. This philosophy argues that monitoring is an important function (it certainly is) and if you are still having resource problems after you have increased your time interval (say, to more than five minutes), you should buy more hardware. If you agree with the latter philosophy, the objects you should monitor are as follows:

- Memory
- Physical disk
- Process
- Processor
- Network interface

- SQLServer:Access Methods
- SQLServer:Buffer Manager
- SQLServer:Databases
- SQLServer:General Statistics
- SQLServer:Latches
- SQLServer:Locks
- SQLServer:Memory Manager
- SQLServer:SQL Statistics
- SQLServer:User Settable

SQL Server 2005 Books Online recommends that you perform your monitoring on a monitoring server rather than on the SQL Server 2005 production server that you want to monitor. In practice, this philosophy is debatable. If the Performance tools run on a remote system, Distributed Component Object Model (DCOM) calls have (arguably) more impact on network resources than does logging all counters directly on the production server.

Exam Tip If an exam question asks whether you should log performance counters locally or remotely, the official Microsoft answer, according to Books Online, is to use remote monitoring.

Using Performance Counters

Whether you decide to monitor only specific counters or all counters associated with specific objects, some counters give you a better insight into SQL Server operation than others. Every DBA has his or her own set of useful counters. You can specify counters that monitor server health, server availability, and SQL Server service availability.

Monitoring Server Health

You can use the following counters to monitor server health. This list is by no means exclusive:

- **Memory: Available Bytes** Shows the available amount of physical memory on the server. An acceptable output for this counter can vary widely based on how much physical memory is in the machine. If you have 2 GB of RAM, it is common for SQL Server 2005 to use 1.7 GB. If no other processes are running on your

SQL Server, ensure you have at least 80 MB available for the Windows operating system. If the value in this counter falls below that amount, consider adding RAM.

- **Memory: Pages/sec** Shows the number of pages that are read from or written to disk. If this counter averages 20, and no other processes are running on the SQL Server 2005 server, you should again consider adding RAM.
- **Network Interface: Bytes total/sec** Shows the amount of traffic through your network interface in bytes per second. You need to compare the value in this counter with baseline values. A sudden rise in the value could indicate your server is suffering an external attack. A sudden fall could indicate connectivity problems.
- **Network Interface: Output Queue Length** Returns the length of an output packet queue in packets. A sustained value higher than 2 indicates network congestion.
- **Paging File: % Usage** Shows the percentage of the page file that is being utilized. If you see more than 70 percent of your page file being utilized, consider adding more RAM for your server.
- **PhysicalDisk: % Disk Time** Measures hard disk activity. If this counter returns an average above 70 percent, you might have hard disk problems. Hard disk problems can, however, be caused by a shortage of RAM, so further investigation is required.
- **LogicalDisk: % Free Space** Indicates when a logical disk volume—typically a disk array—is running out of space. This is always important information, but it is particularly so when the volume stores the database transaction logs.
- **Processor: % Processor Time** Measures CPU activity. If you have a multiprocessor computer, each CPU is an instance of this counter, or you can specify all CPUs as an instance. An average value of 85 percent indicates a serious problem, but if you consistently detect average values greater than 60 percent you should consider upgrading your processors.
- **Processor: % Privileged Time** Returns the percentage of time the processor spends on execution of Microsoft Windows kernel commands. If the value in this counter is consistently high when the values in the Physical Disk counters are high, or if it is increasing through time, you should consider installing a faster or more efficient disk subsystem. Efficient disk subsystems use less privileged time, leaving more processing time available for user applications and hence increasing overall throughput.

- **Process: % Processor Time** Measures CPU activity related to a process. Each process is an instance of this counter. The counter is useful if you suspect that a particular process is using an excessive amount of CPU resources.
- **System: Processor Queue Length** Indicates the number of queued threads that are waiting to be processed. If this exceeds two per CPU for a significant length of time, you potentially have a processor bottleneck.

Monitoring Server Availability

You can use the following counters to monitor server availability. Again, this list is by no means exclusive:

- **SQLServer:Access Methods: Full Scans/sec** Indicates how many full table or index scans are occurring per second. If this number is significantly higher than the baseline value, application performance might be slow.
- **SQLServer:Buffer Manager: Buffer Cache Hit Ratio** Shows the ratio of pages stored in memory rather than on hard disk. This should be as near to 100 percent as possible, but typically anything above 90 percent is acceptable. A value of 90 percent or below might indicate that SQL Server operation is limited by memory constraints.
- **SQLServer:Databases: Transactions/sec** Shows the number of transactions on a given database or on the entire SQL Server per second. If this number is significantly more than its baseline value, there could be issues with your server or database activity.
- **SQLServer:General Statistics: User Connections** Shows the number of user connections to a SQL Server. If this number is significantly more than its baseline value, there could be a deterioration in SQL Server performance as a result of increased user activity.

Monitoring Database Availability and Transaction Log Usage

You can use the following counters to determine the availability of a specific database. If the database is running short of space for transaction logs, this affects availability, and counters that monitor transaction logs are included in this list. As before, the list is not exclusive:

- **SQLServer:Databases: Percent Log Used** Percentage of space in the transaction log that is in use.

- **SQLServer:Databases: Log Growths** Total number of times the transaction log for the database has been expanded.
- **SQLServer:Databases:Data File(s) Size (KB)** Cumulative size (in kilobytes) of all the data files in the database, including any automatic growth.
- **SQLServer:SQL Errors: Errors/sec** Returns the number of SQL errors per second. This information is significant with regard to database availability because it can include errors that cause SQL Server to take the current database offline, errors that cause SQL Server to close the current connection, user errors, and information related to error messages that provide information to users.

Monitoring SQL Availability

You can use the following counters to determine the availability of the SQL Server service. Once again, this list is not exclusive:

- **SQLServer:Latches: Average Latch Wait Time (ms)** Shows the average time for a latch to wait before a request is met. If this number is significantly more than its baseline value, concurrency issues could exist.
- **SQLServer:Locks: Lock Waits/sec** Shows the number of locks per second that could not be satisfied immediately and had to wait for resources. If this number is significantly more than its baseline value, concurrency issues could exist.
- **SQLServer:Locks: Lock Timeouts/sec** Shows the number of locks per second that timed out. If the value in this counter is more than zero, users might be experiencing problems with queries that are not completing.
- **SQLServer:Locks: Number of Deadlocks/sec** Shows the number of locks per second on the SQL Server that become deadlocks. If the value in this counter is more than zero, users might be experiencing problems with queries that are not completing, and applications might be failing.
- **SQLServer:Memory Manager: Total Server Memory (KB)** Shows the amount of memory that is allocated to SQL Server. If this memory is equal to the amount of total physical memory on the machine, you could be experiencing contention because the Windows operating system has not been allocated any RAM to perform its normal operations.
- **SQLServer:SQL Statistics: SQL Re-Compilations/sec** Shows the amount of SQL recompiles per second. If this number is significantly higher than its baseline value, stored procedure execution plans might not be caching appropriately.

- **SQLServer: User Settable: Query** Provides up to 10 customizable counters that you can implement using the `sp_user_counterx` stored procedure (where `x` is a number from 1 through 10). These counters can be used to track customized tasks. For example, if you have a table that holds items in a queue, a user-settable query counter can indicate when the count in the queue table becomes unacceptably high. You can use the counter to trigger an alert that could send you an e-mail or correct the situation (or both). Note that updating this counter frequently can slow down your server.

Exam Tip A very large number of performance counters exist, and nobody could reasonably expect you to memorize all of them. Examination questions typically focus on the most widely used counters, such as those listed in this chapter. If you see a counter you have never heard of before in an exam question, it is probably not the correct answer.

NOTE The sysperfinfo system table

Data for the SQL Server counters is stored in the `sysperfinfo` system table in the master database. Only the first 99 databases are stored in the table.

Using Event Logs

SQL Server 2005 logs system events and user-defined events to the SQL Server error log and the Windows application event log. Both logs automatically *timestamp* all recorded events. Lesson 2 of this chapter discusses the use of information in the SQL Server error log to troubleshoot problems related to SQL Server.

The application event log records events that are logged by applications, and it provides an overall picture of events that occur on the Windows operating system, in addition to events logged by SQL Server and SQL Server Agent. For example, a database application might record a file error in the application log. You can use the Windows Event Viewer to view the Windows application log and to filter the information for events, such as information, warning, error, success audit, and failure audit.

To view the Windows application log, you start Event Viewer from Administrative Tools on the Programs or All Programs menu, and then select Application. SQL Server events are identified by the entry `MSSQLSERVER`. Named instances are identified by `MSSQL$<instance_name>` in the Source column. SQL Server Agent events are identified by the entry `SQLSERVERAGENT`. Named instances of SQL Server Agent events are identified by `SQLAgent$<instance_name>`. Microsoft Search service events are

identified by the entry “Microsoft Search”. To view more information about an event, double-click the event.

If you want to display only SQL Server events, choose Filter from the View menu and select MSSQLSERVER from the Event Source drop-down list. To view only SQL Server Agent events, select SQLSERVERAGENT from the Event Source drop-down list. You can also view the log of a remote computer. To do this, right-click Event Viewer, choose Connect To Another Computer, identify the computer in the Select Computer dialog box, and then click OK.

Quick Check

- Which SQL Server object counter can specifically indicate whether SQL Server operation is limited by memory constraints?

Quick Check Answer

- SQLServer:Buffer Manager: Buffer Cache Hit Ratio. Other memory-based counters—for example, Memory: Pages/sec and Memory: Available Bytes—indicate general memory pressure. SQLServer:Buffer Manager: Buffer Cache Hit Ratio is specific to SQL Server.

Monitoring in a Clustered Environment

If you are monitoring a SQL Server 2005 failover cluster, you need to bear in mind that the server cluster consists of a virtual server that runs under the MSCS service. Microsoft virtual server-based nodes that provide the Microsoft server cluster can host the virtual server. If problems exist on the nodes that host the server cluster, those problems might manifest themselves as issues with your virtual server. To monitor an MSCS virtual server, you need to review the Windows system and application event logs, and the cluster log.

MSCS clusters log errors and events to the system event log. You can also turn on and configure verbose logging for the cluster service to a text file named Cluster.log. By default, Windows Server 2003 (and Windows 2000 Server) enables the cluster log. The recommended default path is %SystemRoot%\Cluster\Cluster.log.

MORE INFO Cluster logging

For more information, access the Microsoft article, “How to turn on cluster logging in Microsoft Cluster Server” at support.microsoft.com/kb/168801/en-us.

If the hardware, operating system, network, and MSCS service are problem-free, you can monitor the MSCS virtual server using the same techniques, performance counters, and traces that you would use for a normal SQL Server 2005 server. The results you obtain are for the active node of the MSCS cluster.

Defining Traces

You can define and create SQL traces by using SQL Server Profiler or Transact-SQL system stored procedures—for example, *sp_trace_create*, *sp_trace_setevent*, *sp_trace_setfilter*, and *sp_trace_setstatus*. Chapters 1 and 2 discuss this in depth, and only a brief summary is included here. As with performance counters, you can use SQL traces both to troubleshoot a problem and to monitor server resources and SQL procedures. A trace can indicate event classes that have high disk input/output (I/O) or CPU usage, or that take a long time to complete. A series of traces recorded over time can indicate increasing disk I/O or CPU usage and that the duration of particular events is increasing.

Using Profiler

Microsoft SQL Server Profiler is a graphical user interface that enables you to trace events that occur in SQL Server 2005 and to capture and save data about each event class to a file or table for analysis. An event class describes actions generated within an instance of SQL Server database engine. Examples of these include the following:

- Login connections, failures, and disconnections
- Transact-SQL SELECT, INSERT, UPDATE, and DELETE statements
- The start or end of a SQL batch
- An error written to the SQL Server error log
- A lock acquired or released on a database object

Profiler displays the data generated by an event in a single row, intersected by information columns that describe the event in detail—for example, *EventClass*, *EventCategory*, *DataColumn*, and so on. Trace data can be saved, or used immediately for analysis. You can replay traces at a later date, although certain events, such as Exception events, are never replayed. You can also save the trace as a template to build similar traces in the future.

MORE INFO SQL Server event classes

For more information about the event classes you can specify in a Profiler trace, search for "SQL Server Event Class Reference" in Books Online or access [msdn2.microsoft.com/en-us/library/ms175481\(d=ide\).aspx](http://msdn2.microsoft.com/en-us/library/ms175481(d=ide).aspx).

Monitoring Users

In addition to monitoring the resources required to run queries against databases, you might need to find out which users are generating specific ad hoc queries. Suppose, for example, a number of users routinely run ad hoc queries against a database on a SQL Server 2005 server, and the server performance is periodically slow. You suspect that one of the users is running badly formed queries, and you need to find out who is causing the problem. In this case, you could either run SQL Trace system stored procedures to trace database activity or use Profiler to create a trace that uses a predefined template. Both methods enable you to compare the login name of the user with the duration and resource usage of the query.

Using Trace Flags

You use trace flags to temporarily set specific server characteristics or to switch off a particular behavior. You can use them to diagnose performance issues or to debug stored procedures. Two types of trace flags exist in SQL Server 2005: session and global. Session trace flags are active for a connection and are visible only to that connection. Global trace flags are set at the server level and are visible to every connection on the server. Some flags can be enabled only as global, and some can be enabled as either global or session.

A global trace flag must be enabled globally. Otherwise, the trace flag has no effect. If a trace flag has either global or session scope, it can be enabled with the appropriate scope. A trace flag that is enabled at the session level never affects another session, and the effect of the trace flag is lost when the server process identifier (SPID) that opened the session logs out.

Table 5-1 lists and describes the trace flags that are available in SQL Server 2005. Trace flag behavior might not be supported in future releases of SQL Server.

Table 5-1 Trace Flags

Trace Flag	Description	Scope
260	Prints versioning information about extended stored procedure dynamic-link libraries (DLLs)	Global or session
1204	Returns the resources and types of locks participating in a deadlock and also the current command affected	Global only
1211	Disables lock escalation based on memory pressure, or based on number of locks	Global or session
1222	Returns the resources and types of locks that are participating in a deadlock and also the current command affected	Global only
1224	Disables lock escalation based on the number of locks—however, memory pressure can still activate lock escalation	Global only
4616	Makes server-level metadata visible to application roles	Global only
2528	Disables parallel checking of objects by DBCC CHECKDB, DBCC CHECKFILEGROUP, and DBCC CHECKTABLE	Global or session
3205	Disables hardware compression for tape drivers	Global or session
3625	Limits the amount of information returned in error messages	Global only
7806	Enables a dedicated administrator connection (DAC) on SQL Server Express	Global only

If you set both trace flag 1211 and 1224, 1211 takes precedence over 1224. However, because trace flag 1211 prevents escalation in every case, even under memory pressure, Microsoft recommends that you use 1224. This approach helps avoid “out-of-locks” errors when many locks are being used.

You can turn a trace flag on or off by using the DBCC TRACEON and DBCC TRACEOFF Transact-SQL commands. To enable the trace flag globally, you use DBCC TRACEON with the -1 argument. To turn off a global trace flag, you use DBCC TRACEOFF with the -1 argument. The DBCC TRACESTATUS command determines which trace flags are currently active.

The following example turns trace flag 3205 on and enables it globally:

```
DBCC TRACEON (3205,-1)
```

Setting Alerts

You can use the Windows Performance Logs and Alerts tool to create an alert that is triggered when a threshold value for a performance counter is reached. The alert can send a message and start an application—for example, a custom application or transaction log backup.

In Performance Logs and Alerts, you right-click Alerts and then choose New Alert Settings. You then name the new alert, add a comment (if appropriate), and add one or more counters. In the New Alert dialog box, you select either Over or Under and then enter a threshold value in the Limit text box. You set the sampling frequency, specify the actions that you want to occur every time the alert is triggered, and set the start and stop schedule for the alert scan.

You can also define alerts by using SSMS and SQL Server Agent. Events generated by SQL Server are entered into the Windows application log. SQL Server Agent reads the application log and compares these events to alerts that you define. When SQL Server Agent finds a match, it fires an alert. In addition to monitoring SQL Server events, SQL Server Agent can also monitor performance conditions and Windows Management Instrumentation (WMI) events.

To define an alert in SQL Server Agent (accessed from Object Explorer in SSMS), you specify the name of the alert, the event or performance condition that triggers the alert, and the action that SQL Server Agent takes in response to the event or performance condition.

Specifying a SQL Server Event

You can specify an alert to occur in response to one or more events. You can use the following parameters to specify the events that trigger an alert:

- **Error number** Specifying this parameter causes SQL Server Agent to fire an alert when a certain error occurs. You need to specify a valid error number and event.

- **Severity level** If you use this parameter, SQL Server Agent fires an alert when any error of a specified severity occurs. For example, you might specify a severity level of 10 to respond to syntax errors in Transact-SQL statements.
- **Database** You can specify that SQL Server Agent fires an alert only when an event with a particular error number or severity occurs in a particular database. If, for example, a SQL Server instance contains a production database and an employees database, you can define an alert that responds to syntax errors in the production database only.
- **Event text** You can specify that SQL Server Agent fires an alert when the specified event contains a particular text string in the event message—for example, the name of a particular table or constraint.

Specifying a Performance Condition

You can specify an alert to occur in response to a particular performance condition. In this case, you specify the performance counter to monitor, a threshold for the alert, and the behavior that the counter must exhibit if the alert is to occur. To set a performance condition, you must define the following items on the SQL Server Agent General page of the New Alert or Alert Properties dialog box:

- **Object** The area of performance to be monitored.
- **Counter** An attribute of the area to be monitored.
- **Instance** The specific instance (if any) of the attribute to be monitored.
- **Alert if counter/Value** The threshold for the alert and the behavior that produces the alert. The behavior is one of the following: Falls Below, Becomes Equal To, or Rises Above a number specified for Value. Value is a threshold number that describes the performance condition counter. For example, to set an alert to occur for the performance object SQLServer:Locks when the Lock Wait Time exceeds 30 minutes, you would choose Rises Above and specify 30 as the value.

MORE INFO Defining alerts

For more information, search for “Defining Alerts” in Books Online or access [msdn2.microsoft.com/en-us/library/ms180982\(d=ide\).aspx](https://msdn2.microsoft.com/en-us/library/ms180982(d=ide).aspx).

Setting Event Notifications

Event notifications execute in response to a variety of Transact-SQL data definition language (DDL) statements and SQL Trace events by sending information about these events to Service Broker. They can be used to log and review changes or activity

occurring on the database, and to perform an action in response to an event in an asynchronous manner instead of a synchronous manner. They run asynchronously, outside the scope of a transaction. Therefore, you can use event notifications inside a database application to respond to events without using any resources defined by the immediate transaction. Unlike SQL Trace, you can use event notifications to perform an action inside an instance of SQL Server in response to a SQL trace event.

When an event notification is created, one or more Service Broker conversations open between an instance of SQL Server and the target service you specify. The conversations typically remain open as long as the event notification exists as an object on the server instance. Every event notification has its own exclusive conversations. Ending a conversation explicitly prevents the target service from receiving more messages, and the conversation will not reopen the next time the event notification fires. Event information is delivered to Service Broker as a variable of type xml that provides information about when an event occurs, the database object affected, the Transact-SQL batch statement involved, and other information. Lesson 2 of this chapter describes Service Broker in more detail.

MORE INFO CREATE EVENT NOTIFICATION

You need to use Transact-SQL statements, in particular the CREATE EVENT NOTIFICATION statement, to create and configure event notifications. For more information, search Books Online for "CREATE EVENT NOTIFICATION" or access [http://msdn2.microsoft.com/en-us/library/ms189453\(SQL.90,d=ide\).aspx](http://msdn2.microsoft.com/en-us/library/ms189453(SQL.90,d=ide).aspx).

Using Event Notifications

You can use SQL Server event classes to configure event notifications. For example, the event classes in the Security Audit event category let you set notifications for connection and disconnection events; backup and restore events; when services start, stop, and pause; and any changes in object permissions, such as the use of the EXECUTE AS option with stored procedures and other commands. If, for example, you want to receive a notification by e-mail when the SQL Server or SQL Server Agent service stops, starts, or pauses, you create an event notification on the Audit Server Starts And Stops event class; if you want to be notified when EXECUTE AS is used, you create an event notification on the Audit Database Principal Impersonation event class; if you want to be notified when a CREATE, ALTER, or DROP statement is

executed on a database object, you create an event notification on the Audit Database Object Management event class.

MORE INFO Security audit event category

For more information, search for "Security Audit Event Category (SQL Server Profiler)" in Books Online or access [msdn2.microsoft.com/en-us/library/ms191148\(d=ide\).aspx](https://msdn2.microsoft.com/en-us/library/ms191148(d=ide).aspx).

PRACTICE **Setting an Alert**

In Chapter 1, you set up and used a counter log and generated a trace by using Profiler. These practices are relevant to this chapter also, and you should review them. In this practice session, you set up an alert that writes an event into the Windows application event log when the Processor: %Processor Time counter reaches a value greater than 50 percent. This happens frequently, so it should be easy to test this alert. In the real world, you would set a more useful alert that sends a message or starts a program if (for example) a logical volume or a transaction log is approaching its capacity.

► Practice: Setting a Performance Alert

In this practice, you configure an alert and add the Process: %Processor Time counter. In a production environment, you would configure this alert to run continuously and set the counter sampling intervals accordingly. For convenience, this practice configures the alert with a sampling period of one second and specifies that you start and stop it manually.

1. Log in to your domain at your member server by using either your domain administrator account or an account that has been added to the sysadmin server role. If you are using virtual machine software, log in to your domain and connect to your member server.
2. From the Programs (or All Programs) menu, choose Administrative Tools, choose Performance, and expand Performance Logs And Alerts.
3. Right-click Alerts and choose New Alert Settings, as shown in Figure 5-1.

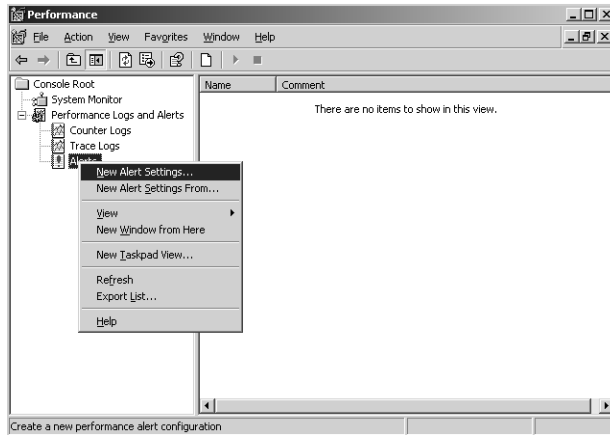


Figure 5-1 Selecting new alert settings.

4. Name the alert **Processor_Time_Alert**. Click Ok.
5. On the General tab of the Processor_Time_Alert Properties dialog box, click Add.
6. In the Add Counters dialog box, shown in Figure 5-2, indicate that you are adding counters from your member server by selecting the Select Counters From Computer option. If you have followed the setup advised in this book, your server is named \\GLASGOW rather than \\OFFICE as shown in the figure. This makes no difference whatsoever to the procedure.



Figure 5-2 The add counters dialog box.

7. If necessary, select Processor from the Performance Object drop-down list, the %Processor Time counter, and the 0 instance. Click Add. Click Close.

- On the General tab, choose to alert when the value is over 50 and to sample data every 1 second, as shown in Figure 5-3. Click Apply.

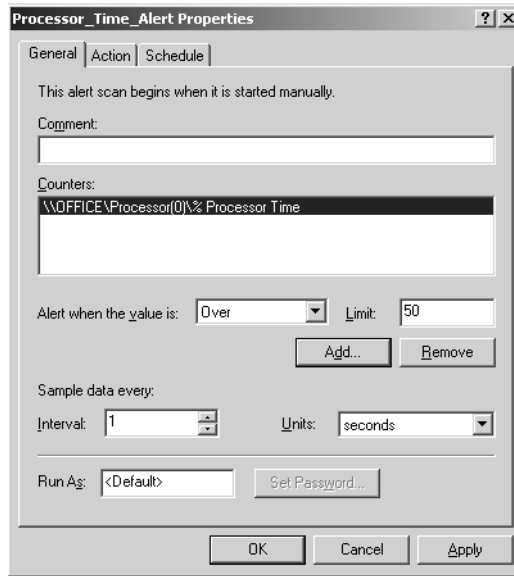


Figure 5-3 Specifying the alert.

- On the Action tab, verify that the Log An Entry In The Application Event Log check box is selected, as shown in Figure 5-4. If necessary, click Apply.

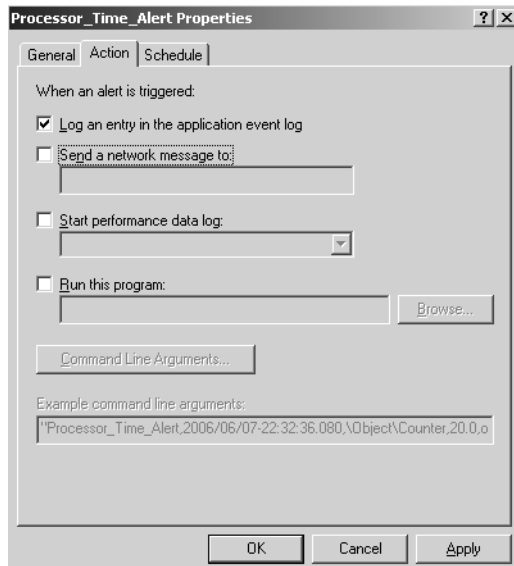


Figure 5-4 Specifying the action.

10. On the Schedule tab, configure the alert to stop and start manually, as shown in Figure 5-5.

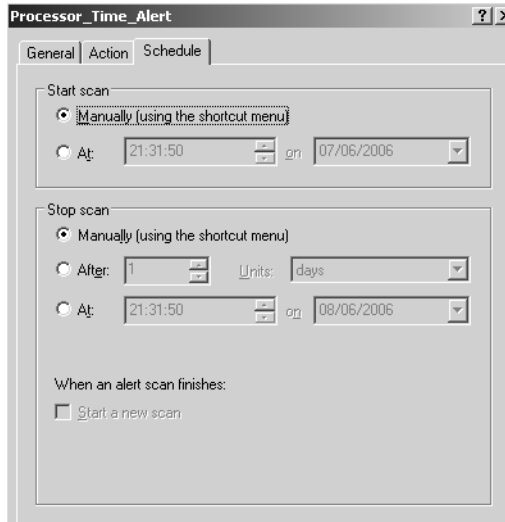


Figure 5-5 Specifying the schedule.

11. Click Ok. In the Performance console, right-click the alert and choose Start, as shown in Figure 5-6.

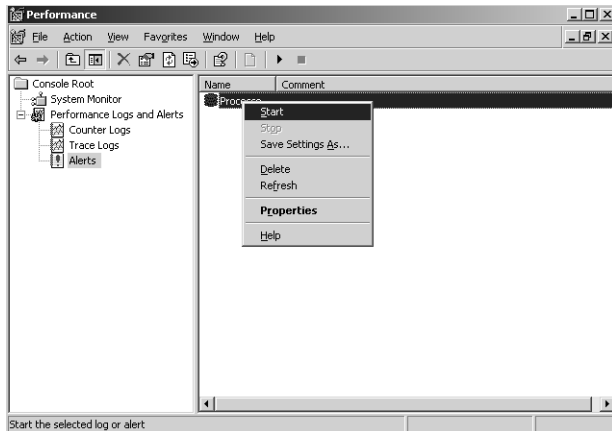


Figure 5-6 Starting the alert.

12. Open a major application—for example, SQL Server Profiler or the Database Engine Tuning Advisor.

- From the Administrative Tools menu, choose Event Viewer, select the Application log, and double-click a Sysmonlog event to access its event properties, as shown in Figure 5-7.

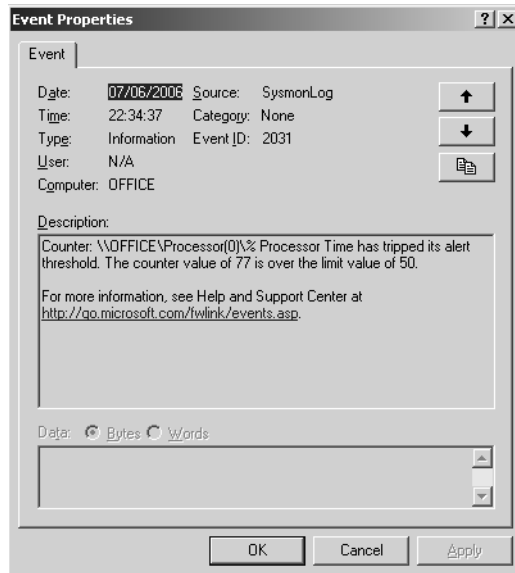


Figure 5-7 The alert event.

- In the Performance console, right-click the alert and choose Stop.

Lesson Summary

- Monitoring involves taking periodic snapshots of current performance to isolate processes that could cause problems, and gathering information continuously over time to track performance trends.
- To determine your monitoring strategy and set monitoring thresholds, you need to clearly define your goals, specify your baseline conditions, and select the appropriate components to monitor and the tools with which to monitor them.
- To determine whether your SQL Server system is performing optimally, you need to take performance measurements at regular intervals over time to establish a server performance baseline and track trends.
- You should monitor server health, server availability, database availability, transaction log usage, and SQL availability.
- You can define and create SQL traces by using SQL Server Profiler or Transact-SQL system stored procedures.

- You can use the Windows Performance Logs and Alerts tool to create an alert that is triggered when a threshold value for a performance counter is reached. The alert can send a message and start an application.
- Event notifications execute in response to a variety of Transact-SQL DDL statements and SQL Trace events by sending information about these events to Service Broker.

Lesson Review

You can use the following questions to test your knowledge of the information in Lesson 1, “Defining and Implementing Monitoring Standards for a Physical Server.” The questions are also available on the companion CD if you prefer to review them in electronic form.

NOTE Answers

Answers to these questions and explanations of why each answer choice is right or wrong are located in the “Answers” section at the end of the book.

1. You want to receive an e-mail message when the SQL Server service and the SQL Server Agent service are started, stopped, or paused. What should you do?
 - A. Use Profiler to create a trace that includes tracking the Audit Server Starts And Stops event class.
 - B. Create an event notification on the Audit Server Starts And Stops event class.
 - C. Use Profiler to create a trace that includes the Audit Server Principal Impersonation event class.
 - D. Create an event notification on the Audit Server Principal Impersonation event class.
2. All transaction logs on a SQL Server 2005 server have been created without file growth enabled. You need to ensure that you are notified when a transaction log is running out of space. What should you do?
 - A. Create a performance alert on the SQLServer:Databases: Percent Log Used counter on each transaction log. Configure the alerts to notify you when the counter value is greater than 75 percent.
 - B. Create a performance alert on the SQLServer:Databases: Percent Log Used counter on each transaction log. Configure the alerts to notify you when the counter value is less than 25 percent.

- C. Create a performance alert on the LogicalDisk: % Free Space counter on the logical volume that stores the transaction logs. Have the alert notify you when the value in this counter is more than 75 percent.
 - D. Create a performance alert on the LogicalDisk: % Free Space counter on the logical volume that stores the transaction logs. Have the alert notify you when the value in this counter is less than 25 percent.
3. You are a DBA at Litware, Inc. A number of applications developers run ad hoc queries, stored procedures, and applications against a database on one of your SQL Server 2005 servers. Some of this activity is stressing the server resources badly and degrading the machine's performance. You want to find out which transactions are stressing the server, and which of the developers is initiating the transactions. How can you find this out? (Choose all that apply).
- A. Implement an alert that triggers on a high value in the SQLServer:Access Methods: Full Scans/sec performance counter. Configure the alert to start an application that captures the login identities of all currently connected users.
 - B. Create an event notification on the Audit Database Object Management event class, and configure the notification to alert you by e-mail.
 - C. Run system stored procedures to create a SQL trace that gathers database activity.
 - D. Use Profiler to create a SQL trace that uses a predefined template.
4. You have been capturing performance logs on a SQL Server 2005 server over a period of time. You notice that CPU usage has been increasing steadily and is currently averaging almost 80 percent. The server has a single CPU. You want to find out whether a specific application or service is causing a problem, or whether you simply require more CPU resource. If a single service or application is stressing the CPU, you want to identify it. What should you add to your performance log?
- A. The `_total` instance of Processor: % Privileged Time
 - B. Each instance of Processor: % Privileged Time
 - C. The `_total` instance of Process: % Processor Time
 - D. Each instance of Process: % Processor Time
5. You need to devise a monitoring strategy that enables you to make decisions about server performance history. Your monitoring strategy should also advise you on how to best allocate funds for hardware upgrades. What should you do?

- A. Create a series of performance alerts to notify you when Memory: Available Bytes, Memory: Pages/sec, Network Interface: Bytes total/sec, Network Interface: Output Queue Length, Paging File: % Usage, Physical Disk: % Disk Time, LogicalDisk: % Free Space, Processor: % Processor Time, Processor: % Privileged Time, and System: Processor Queue Length exceed or fall below predefined thresholds.
 - B. Use System Monitor in graph mode to display information about Memory: Available Bytes, Memory: Pages/sec, Network Interface: Bytes total/sec, Network Interface: Output Queue Length, Paging File: % Usage, Physical Disk: % Disk Time, LogicalDisk: % Free Space, Processor: % Processor Time, Processor: % Privileged Time, and System: Processor Queue Length. Perform this task at regular intervals during periods of peak server activity.
 - C. Configure a counter log to capture information about Memory: Available Bytes, Memory: Pages/sec, Network Interface: Bytes total/sec, Network Interface: Output Queue Length, Paging File: % Usage, Physical Disk: % Disk Time, LogicalDisk: % Free Space, Processor: % Processor Time, Processor: % Privileged Time, and System: Processor Queue Length. Schedule the log to execute at regular intervals, and store the information in a database.
 - D. Use Task Manager to capture information to a SQL Server database table. Perform this task at regular intervals during periods of peak server activity.
6. Developers are running new applications against a table in the HumanResources database. Some of these applications result in deadlocks. You want to find out what types of locks are participating in a deadlock and also the current command affected. You set trace flag 1204 locally. This has no effect. What should you have done?
- A. Set trace flag 1222 locally.
 - B. Set trace flag 1204 globally.
 - C. Monitor the SQLServer:Locks: Number of Deadlocks/sec performance counter.
 - D. Monitor the SQLServer:Locks: Lock Waits/sec performance counter.

Lesson 2: Choosing the Appropriate Information to Monitor

Lesson 1 discussed whether to monitor all instances of all counters provided by selected performance objects, or whether to include only specified counters and instances in your baseline figures and performance log statistics. Even if you choose to collect as much data as possible, you do not want to inspect all of it for every problem you encounter. If you follow this philosophy, you need to store your data in a database, and you need an application to analyze the results.

This lesson discusses various methods of collecting information to help you decide what resources are coming under stress and what monitoring information you should examine. DMVs, introduced in SQL Server 2005, return server state information that you can use to monitor the health of a server instance, diagnose problems, and tune performance. This lesson discusses the *sys.dm_exec_query_stats* DMV, which you can use to obtain information about query plan execution. The SQL Server log, commonly used in conjunction with the Windows event logs, can detect current or potential problem areas and enables you to view automatic recovery messages. The lesson discusses how you can view and recycle that log.

After this lesson, you will be able to:

- Collect information to help you determine what resources are coming under stress and what monitoring information you should examine.
- Use DMVs that return server state information to assist you in obtaining performance statistics.
- Access information in the SQL Server and SQL Server Agent logs.
- Analyze waits.
- Check the availability and status of services.

Estimated lesson time: 60 minutes

Using the *sys.dm_exec_query_stats* DMV

The *sys.dm_exec_query_stats* DMV is a server-scoped DMV that returns aggregate performance statistics for cached query plans. The DMV contains one row per query plan, and the lifetime of the row is tied to the plan itself. When SQL Server removes a plan from the cache, it eliminates the corresponding row from the view. The DMV returns the following aggregate performance statistics for cached query plans:

- The number of times SQL Server has recompiled a query plan.

- A *handle* to the query plan. If the query was executed as part of a batch, this is a binary hash of the batch's text. If the query is part of a stored procedure, this value—together with *statement_start_offset* and *statement_end_offset* (which are described later in this section)—can be used to retrieve the SQL text of the query by calling the *sys.dm_exec_sql_text* dynamic management function.
- The last time the query was executed.
- The number of times the query has been executed.
- The total amount of the following resources consumed by all invocations of the query plan as well as the least and greatest amount of CPU consumed by a single invocation of the query plan:
 - CPU
 - Physical Reads
 - Logical Writes
 - Logical Reads
 - CLR Time
 - Elapsed Time

Querying the *sys.dm_exec_query_stats* DMV

The *sys.dm_exec_query_stats* DMV returns a large number of columns, which you can select to find specific information about query execution:

- **sql_handle** If the query was executed as part of a batch, this is a binary hash of the batch's text. If the query is part of a stored procedure, this value—together with *statement_start_offset* and *statement_end_offset*—can be used to retrieve the SQL text of the query by calling the *sys.dm_exec_sql_text* dynamic management function.
- **statement_start_offset** Indicates, in bytes, the starting position of the query that the row describes within the text of its batch or persisted object.
- **statement_end_offset** Indicates, in bytes, the ending position of the query that the row describes within the text of its batch or persisted object.
- **plan_generation_num** Returns the number of times this plan has been recompiled while it remained in the cache.
- **plan_handle** Returns a pointer to the plan. This value can be passed to the *dm_exec_query_plan* dynamic management function.
- **creation_time** Returns the time at which the plan was compiled.

- **last_execution_time** Returns the last time at which the plan was executed.
- **execution_count** Returns the number of times that the plan has been executed since it was last compiled.
- **total_worker_time** Returns the total amount of CPU time in microseconds that was consumed by executions of this plan since it was compiled.
- **last_worker_time** Returns the CPU time in microseconds that was consumed the last time the plan was executed.
- **min_worker_time** Returns the minimum CPU time in microseconds that this plan has ever consumed during a single execution.
- **max_worker_time** Returns the maximum CPU time in microseconds that this plan has ever consumed during a single execution.
- **total_physical_reads** Returns the total number of physical reads performed by executions of this plan since it was compiled.
- **last_physical_reads** Returns the number of physical reads performed the last time the plan was executed.
- **min_physical_reads** Returns the minimum number of physical reads that this plan has ever performed during a single execution.
- **max_physical_reads** Returns the maximum number of physical reads that this plan has ever performed during a single execution.
- **total_logical_writes** Returns the total number of logical writes performed by executions of this plan since it was compiled.
- **last_logical_writes** Returns the number of logical writes performed the last time the plan was executed.
- **min_logical_writes** Returns the minimum number of logical writes that this plan has ever performed during a single execution.
- **max_logical_writes** Returns the maximum number of logical writes that this plan has ever performed during a single execution.
- **total_logical_reads** Returns the total number of logical reads performed by executions of this plan since it was compiled.
- **last_logical_reads** Returns the number of logical reads performed the last time the plan was executed.
- **min_logical_reads** Returns the minimum number of logical reads that this plan has ever performed during a single execution.
- **max_logical_reads** Returns the maximum number of logical reads that this plan has ever performed during a single execution.

- **total_clr_time** Returns the time in microseconds consumed inside .NET common language runtime (CLR) objects by executions of this plan since it was compiled. The CLR objects can be stored procedures, functions, triggers, types, and aggregates.
- **last_clr_time** Returns the time consumed by execution inside .NET common language runtime (CLR) objects during the last execution of this plan. The CLR objects can be stored procedures, functions, triggers, types, and aggregates.
- **min_clr_time** Returns the minimum time in microseconds that this plan has ever consumed inside .NET common language runtime (CLR) objects during a single execution. The CLR objects can be stored procedures, functions, triggers, types, and aggregates.
- **max_clr_time** Returns the maximum time in microseconds that this plan has ever consumed inside the .NET common language runtime (CLR) during a single execution. The CLR objects can be stored procedures, functions, triggers, types, and aggregates.
- **total_elapsed_time** Returns the total elapsed time in microseconds for completed executions of this plan.
- **last_elapsed_time** Returns the elapsed time in microseconds for the most recently completed execution of this plan.
- **min_elapsed_time** Returns the minimum elapsed time in microseconds for any completed execution of this plan.
- **max_elapsed_time** Returns the maximum elapsed time in microseconds for any completed execution of this plan.

Exam Tip You should treat the preceding list as a reference. It is unlikely that you will be asked to remember all the columns that can be returned by this DMV. You might, however, be asked about the DMV's function and the type of information it can provide.

The following code block returns information about the top five queries run against the master database by the maximum number of physical reads:

```
USE master;
GO
SELECT TOP 5 sql_handle, last_execution_time, max_physical_reads
AS [max_physical_reads]
FROM sys.dm_exec_query_stats
ORDER BY creation_time DESC;
GO
```

Using the SQL Server Log

You create a new SQL Server error log each time you start an instance of SQL Server. You should view the SQL Server error log to ensure that processes have completed successfully (for example, backup and restore operations, batch commands, and scripts). This can help you to detect any current or potential problem areas, including automatic recovery messages—particularly if an instance of SQL Server has been stopped and restarted—kernel messages, and so on.

NOTE Error Log

The terms *SQL Server log* and *SQL Server error log* are interchangeable.

You can use the *sp_cycle_errorlog* system stored procedure to cycle the error log files without having to restart the instance of SQL Server. Typically, SQL Server retains backups of the previous six logs and gives the most recent log backup the extension .1, the second most recent the extension .2, and so on. The current error log file (ERRORLOG) has no extension. You can view the SQL Server error log by using SSMS or any text editor. By default, the error log file and backup files are located in the `\Program Files\Microsoft SQL Server\MSSQL.1\MSSQL\LOG\` folder.

Comparing Error and Application Log Output

You can use both the SQL Server error log and the Windows application event log to identify the cause of problems. For example, while monitoring the SQL Server error log, you might encounter error messages that do not contain cause information. By comparing the dates and times for events between these logs, you can narrow the list of probable causes. The SSMS Log File Viewer lets you integrate SQL Server, SQL Server Agent, and the Windows logs into a single list, making it easier to correlate related Windows application events and SQL Server events.

Although the Windows application event log is the most useful for displaying SQL Server errors and events, you should not ignore the other event logs on a SQL Server 2005 server. Three logs are available:

- **System log** Records events logged by the Windows operating system components. For example, the failure of a driver or other system component to load during startup is recorded in the system log.
- **Security log** Records security events, such as failed login attempts. This log helps to track changes to the security system and identify possible breaches to

security. For example, attempts to log on to the system might be recorded in the security log, depending on the audit settings. Only members of the sysadmin fixed server role can view the security log on a SQL Server 2005 server.

- **Application log** Records events that are logged by applications in addition to events logged by SQL Server and SQL Server Agent.

Analyzing Waits

Waits can occur when a worker thread is waiting for a resource that has been locked by another process, when a worker thread is idle because it is waiting for a task to be assigned, or when a thread is waiting for an external event to complete. If SQL Server performance is degrading and you want to access wait statistics to determine whether an excessive number of waits is causing a problem, you can access the *sys.dm_os_wait_stats* DMV. Alternatively, if you want to track trends and monitor the number and types of waits on your SQL Server 2005 servers, you should consider adding the SQLServer:Wait Statistics object to the list of objects in your performance log.

Wait Types

Waits can occur for a variety of reasons. The following are typical wait types:

- **Resource wait** Occurs when a thread (or worker) requests access to a resource that is not available because the resource is being used by another worker or is not yet available for some other reason. Examples of resource waits are locks, latches, and network and disk I/O waits. Lock and latch waits are waits on synchronization objects.
- **Queue wait** Occurs when a worker is idle, waiting for work to be assigned. Queue waits are most typically associated with system background tasks such as the deadlock monitor and deleted record cleanup tasks. These tasks wait for work requests to be placed into a work queue. Queue waits might also periodically become active even if no new packets have been put on the queue.
- **External wait** Occurs when a SQL Server worker is waiting for an external event to finish. When you diagnose blocking issues, you need to remember that external waits do not always imply that the worker is idle because the worker might be actively running external code.

Using the *sys.dm_os_wait_stats* DMV

The *sys.dm_os_wait_stats* DMV returns information about waits encountered by threads that are in execution. You can use the view to diagnose performance issues with SQL Server and also with specific queries and batches. The DMV returns the following information:

- **wait_type** Name of the wait type. This can be a resource wait, queue wait, or external wait.
- **waiting_tasks_count** Number of waits on this wait type. This counter is incremented at the start of each wait.
- **wait_time_ms** Total wait time for this wait type in milliseconds. This time is inclusive of *signal_wait_time*.
- **max_wait_time_ms** Maximum wait time on this wait type.
- **signal_wait_time** Difference between the time the waiting thread was signaled and when it started running.

Wait times during query execution can indicate bottlenecks within the query, and server-wide wait counts can indicate bottlenecks in query interactions within the server instance. For example, lock waits indicate data contention by queries, page I/O latch waits indicate slow I/O response times, and page latch update waits indicate incorrect file layout.

All data returned by this DMV is cumulative since the last time the statistics were reset or the server was started. Therefore, if the DMV returns a high value for (say) *max_wait_time_ms*, you need to determine whether this is because of current performance problems. In this case, you need to reset the statistics in the DMV and query the counters again when new statistics have accumulated. You can reset the contents of the DMV by running the following command:

```
DBCC SQLPERF ('sys.dm_os_wait_stats', CLEAR);  
GO
```

Using the *SQLServer:Wait Statistics* Object

The *SQLServer:Wait Statistics* performance object contains performance counters that report information about wait status. You can add this object to your performance log if you consider that waits are or could become a problem. As with the performance objects described in Lesson 1, you can choose to either add the

entire object or to add selected counters and instances. The SQLServer:Wait Statistics performance object contains the following counters:

- **Lock waits** Returns statistics for processes waiting on a lock
- **Log buffer waits** Returns statistics for processes waiting for the log buffer to be available
- **Log write waits** Returns statistics for processes waiting for the log buffer to be written
- **Memory grant queue waits** Returns statistics for processes waiting for a memory grant to become available
- **Network IO waits** Returns statistics relevant to waits on network I/O
- **Non-Page latch waits** Returns statistics relevant to non-page latches
- **Page IO latch waits** Returns statistics relevant to page I/O latches
- **Page latch waits** Returns statistics relevant to page latches, not including I/O latches
- **Thread-safe memory objects waits** Returns statistics for processes waiting on thread-safe memory allocators
- **Transaction ownership waits** Returns statistics relevant to processes synchronizing access to a transaction
- **Wait for the worker** Returns statistics relevant to processes waiting for a worker to become available
- **Workspace synchronization waits** Returns statistics relevant to processes synchronizing access to a workspace

Each counter in the object contains the following instances:

- **Average wait time (ms)** Indicates the average time for the selected type of wait
- **Cumulative wait time (ms) per second** Indicates the aggregated wait time per second for the selected type of wait
- **Waits in progress** Indicates the number of processes currently waiting
- **Waits started per second** Indicates the number of waits started per second of the selected type of wait

Tracing Resource Usage

A SQL trace contains event categories and describes an event class within a category by using data columns. You can generate SQL traces by using system stored procedures, but typically you use SQL Server Profiler for this purpose. Whichever way

you generate a trace, Profiler provides a graphical method of viewing it. You can use Profiler to view a large number of columns in a trace. Typically, the following are the most useful:

- **ApplicationName 1 (column 10)** The name of the client application that created the connection to an instance of SQL Server. This column is populated with the values passed by the application and not with the name of the program.
- **Binary Data (column 2)** A binary value that indicates the event class captured in the trace.
- **ClientProcessID 1 (column 9)** The identity assigned by the host computer to the process where the client application is running. This data column is populated if the client process ID is provided by the client.
- **CPU (column 18)** The amount of CPU time in milliseconds used by the event.
- **DatabaseName (column 35)** The name of the database against which the user statement is running.
- **DBUserName 1 (column 40)** The SQL Server user name of the client.
- **Duration (column 13)** The duration of the event in microseconds.
- **Error (column 31)** The error number of the specified event.
- **EventClass 1 (column 27)** The type of event class that is captured.
- **IntegerData (column 25)** The integer value dependent on the event class captured in the trace.
- **LoginName (column 11)** The name of the login of the user (either the SQL Server security login or the Windows login credentials in the form of DOMAIN\User-name).
- **ObjectID (column 22)** The system-assigned identity of the object.
- **Reads (column 16)** The number of read operations on the logical disk that are performed by the server on behalf of the event. These read operations include all reads from tables and buffers during the statement's execution.
- **ServerName 1 (column 26)** The instance of SQL Server that is being traced.
- **SessionLoginName (column 64)** The login name of the user who originated the session.
- **Severity (column 20)** The severity level of the exception event.
- **SourceDatabaseID (column 62)** The identity of the database in which the source of the object exists.

- **SPID (column 12)** The SPID that is assigned by SQL Server to the process that is associated with the client.
- **TextData (column 1)** The text value dependent on the event class that is captured in the trace.
- **Transaction ID (column 4)** The system-assigned identity of the transaction.
- **Type (column 57)** An integer value that identifies the event class captured in the trace.
- **Writes (column 17)** The number of physical disk write operations that are performed by the server on behalf of the event.

MORE INFO Trace data columns

The list of data columns given here, while substantial, is not inclusive. For a full list, search for “Describing Events by Using Data Columns” in Books Online or access [msdn2.microsoft.com/en-us/library/ms190762\(d=ide\).aspx](https://msdn2.microsoft.com/en-us/library/ms190762(d=ide).aspx).

When analyzing a trace to determine resource usage and the performance of the event class being traced, you want to know the event class and the identity of the SQL Server instance, the database, the process, the transaction, and the user login. However, the columns of most interest to determine resource usage are Reads, Writes, Duration, and CPU.

Event Categories

Profiler enables you to record events as they occur in an instance of the SQL Server database engine. The recorded events are instances of the event classes in the trace definition. In SQL Server Profiler, event classes and their event categories are available on the Events Selection tab of the Trace File Properties dialog box. The event classes available are too numerous to list here. However, the event categories are as follows:

- **Broker** Includes event classes that are produced by the Service Broker
- **CLR** Includes event classes that are produced by the execution of .NET CLR objects
- **Cursors** Includes event classes that are produced by cursor operations
- **Database** Includes event classes that are produced when data or log files grow or shrink automatically

- **Deprecation** Includes deprecation-related events
- **Errors and Warnings** Includes event classes that are produced when a SQL Server error or warning is returned—for example, if an error occurs during the compilation of a stored procedure or an exception occurs in SQL Server
- **Full Text** Includes event classes that are produced when full-text searches are started, interrupted, or stopped
- **Locks** Includes event classes that are produced when a lock is acquired, canceled, released, or has some other action performed on it
- **Objects** Includes event classes that are produced when database objects are created, opened, closed, dropped, or deleted
- **OLEDB** Includes event classes that are produced by object linking and embedding for database (OLE DB) calls
- **Performance** Includes event classes that are produced when SQL data manipulation language (DML) operators execute
- **Progress Report** Includes the Progress Report: Online Index Operation event class
- **Scans** Includes event classes that are produced when tables and indexes are scanned
- **Security Audit** Includes event classes that are used to audit server activity
- **Server** Contains general server events
- **Sessions** Includes event classes produced by clients connecting to and disconnecting from an instance of SQL Server
- **Stored Procedures** Includes event classes produced by the execution of stored procedures
- **Transactions** Includes event classes produced by the execution of Microsoft Distributed Transaction Coordinator transactions or by writing to the transaction log
- **TSQL** Includes event classes produced by the execution of Transact-SQL statements passed to an instance of SQL Server from the client
- **User-Configurable** Includes event classes that you can define

Real World

Ian Mclean

I've included a good number of data columns and event categories in the preceding lists—for completeness and so that you have a reference that should enable you to select the event category and event classes you want to trace, and the columns you want to inspect. I don't spend my life memorizing such lists, and I wouldn't recommend doing so. In the real world, DBAs typically capture traces by using event classes that they find from experience provide the answers for which they are looking, and they inspect the columns they consider relevant. I usually look at event classes in the Security Audit, Locks, Scans, and Errors And Warnings event categories, and I pay particular attention to the Reads, Writes, Duration, CPU, and SPID columns.

Checking Service Availability and Status

A service is a type of application that runs in the system background. Services usually provide core operating system features, such as Web serving, event logging, or file serving, and they can run without showing a user interface on the computer desktop. The SQL Server database engine, SQL Server Agent, and several other SQL Server components run as services. Services typically start when the operating system starts. Some services (for example, the SQL Server Agent service) do not start automatically in a default installation. You can configure such services to start automatically by using the SQL Server Surface Area Configuration Tool for Services and Connections or the Windows Services console.

Lesson 1 describes how you can generate a notification to monitor when a service starts, stops, or pauses. You can also use the SQL Server log and the Windows application event log to monitor services.

Service Broker

Service Broker is not itself a service but is used, for example, to implement notifications that inform you when services have started. Its primary function is to help developers build asynchronous, loosely coupled applications in which independent components work together to accomplish a task. However, application development is beyond the scope of this chapter, and we shall look instead at how Service Broker impinges on applications and services.

Many database applications include tables that function as queues of work to be accomplished as resources permit. Service Broker provides mechanisms for automatically starting programs that process a queue when there is useful work for the program to do, and it provides queuing as an integral part of the database engine.

Service Broker is designed around the basic functions of sending and receiving messages. Each message forms part of a *conversation*, which is a reliable, persistent communication channel. Each message and conversation has a specific type that Service Broker enforces.

An application sends messages to a service, which is a name for a set of related tasks. An application receives messages from a queue, which is a view of an internal table.

Messages for the same task are part of the same conversation. Within each conversation, Service Broker guarantees that an application receives each message exactly once, in the order in which the messages are sent. Certificate-based security helps you protect sensitive messages and control access to services.

The Service Broker framework provides a Transact-SQL interface for sending and receiving messages, combined with a set of guarantees for message delivery and processing. Service Broker guarantees that a program receives each message in a conversation exactly once in the order in which the message is sent, not the order in which the message enters the queue. Integrated queuing means that normal database maintenance and administration also include Service Broker. Typically, a DBA has no routine maintenance tasks related to Service Broker.

The SQL Server Service

SQL Server 2005 runs on the Microsoft Windows 2000 Server or Windows Server 2003 operating systems as the SQL Server service. When you start an instance of the SQL Server database engine, you are starting the SQL Server service. After you start the SQL Server service, users can establish new connections to the server. The SQL Server service can be started and stopped as a service, either locally or remotely. The SQL Server service is referred to as SQL Server (MSSQLSERVER) if it is the default instance, or MSSQL\$<instancename> if it is a named instance.

You can configure a notification to inform you when an instance of the SQL Server service starts, stops, or pauses. The SQL Server log can help you to detect any current or potential problem areas—particularly if an instance of SQL Server has been stopped and restarted. The SQL Server service also writes events into the Windows application event log.

The SQL Server Agent Service

The SQL Server Agent service runs jobs, monitors SQL Server, processes alerts, and allows you to automate administrative tasks. The service must be running before local or multi-server administrative jobs can run automatically.

SQL Server maintains up to nine SQL Server Agent error logs. Each archived log has an extension that indicates the relative age of the log. For example, an extension of .1 indicates the newest archived error log and an extension of .9 indicates the oldest archived error log. The current log has no extension. The SQL Server Agent error log records warnings and errors by default. The log contains warning messages that provide information about potential problems—for example: “Job <job_name> was deleted while it was running,” and error messages that typically require intervention by an administrator, such as “Unable to start mail session.” Error messages can be sent to a specific user or computer by using net send.

By default, execution trace messages are not written to the SQL Server Agent error log because they can fill it. Because the log adds to the server’s processing load, you need to consider carefully whether you should enable the log to capture such messages—typically, only when you are debugging a specific problem.

You need to stop the SQL Server Agent service if you want to modify the location of the SQL Server Agent error log. You can, however, cycle the SQL Server Agent log at any time without stopping SQL Server Agent.

To view the SQL Server Agent error log, connect to an instance of the SQL Server database engine in Object Explorer within SSMS and then expand that instance. Expand SQL Server Agent, expand Error Logs, right-click the error log you want to view, and then choose View Agent Log. In the Select Logs pane, select the check box for a type of logged item. Optionally, you can click Filter and enter parameter values in the Filter Settings dialog box to filter the log contents. Select the Apply Filter check box if you have selected filter parameters, and click OK in the Filter Settings dialog box. You can view the log contents under Log File Summary.

NOTE Empty SQL Server Agent logs.

When the log is empty, you cannot open it.

In addition to using the SQL Server Agent error log, you can configure a notification to inform you when an instance of the SQL Server Agent service starts, stops, or pauses. The SQL Server service also writes events into the Windows application event log.

Microsoft Clustering Service

Failover clustering in Microsoft SQL Server 2005 is implemented by the MSCS service and provides high-availability support for a SQL Server instance. You can configure a SQL Server instance on one node of a failover cluster to fail over to any other node in the cluster during a hardware failure, operating system failure, or planned upgrade.

A failover cluster is a combination—known as a *resource group*—of one or more nodes (servers) with two or more shared disks. A virtual server is the combination of a resource group, its network name, and an Internet protocol (IP) address. It appears on the network as if it were a single computer, but it provides failover from one node to another if the current node becomes unavailable.

A failover cluster appears as a normal application or server, but it has additional functionality that increases its availability. You can consider both the server cluster itself and the SQL Server instance installed on a cluster server virtual servers. In a failover clustering scenario, you can create multiple virtual servers in a cluster, but each virtual server can have only one instance of SQL Server installed.

As you saw in Lesson 1, to monitor the MSCS service you need to review the Windows system and application event logs, and the cluster log on the node that supports the virtual server. The service logs errors and events to the Windows system event log. You can also enable and configure verbose logging for the service to a text file named Cluster.log. By default, Windows Server 2003 (and Windows 2000 Server) enables the cluster log. The recommended default path is %SystemRoot%\Cluster\Cluster.log.

PRACTICE Viewing and Recycling the SQL Error Log

The SQL Server error log contains user-defined events and certain system events. You can use this error log to troubleshoot problems related to SQL Server. A new log starts whenever you start an instance of SQL Server. However, you can start a new log without stopping and restarting SQL Server by using the Transact-SQL *sp_cycle_errorlog* stored procedure. In this practice session, you inspect the current log and then start a new one.

► **Practice 1: Viewing the SQL Server Error Log**

In this practice, you view the SQL Server error log and filter for SPID57.

1. Log in to your domain at your member server by using either your domain administrator account or an account that has been added to the sysadmin server role. If you are using virtual machine software, log in to your domain and connect to your member server.

- From Programs, Microsoft SQL Server 2005, click SQL Server Management Studio. Connect to the database engine on your member server, specifying Windows Authentication. Connect using TCP/IP, and specify the AdventureWorks database.
- In Object Explorer, expand your member server, expand Management, and then expand SQL Server Logs as shown in Figure 5-8. Note that on your screen the server name is GLASGOW rather than OFFICE.

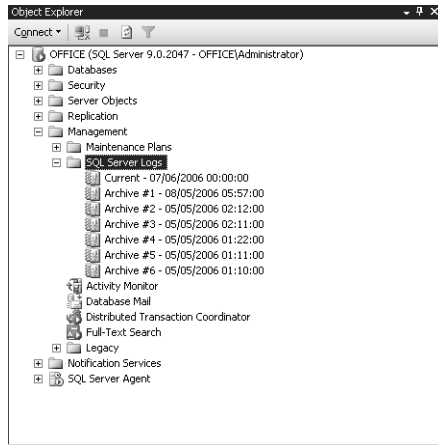


Figure 5-8 Accessing the SQL Server logs.

- Right-click the current log and choose View SQL Server Log. The log appears, as shown in Figure 5-9.

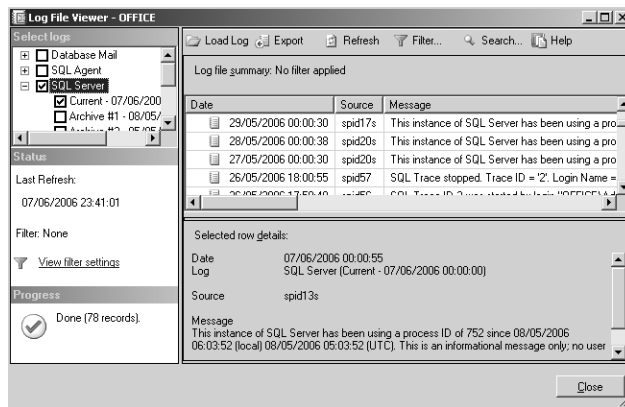


Figure 5-9 The current SQL Server log.

5. Click Filter on the toolbar. In the Filter Settings dialog box, specify the Source as **spid57** and select the Apply Filter check box, as shown in Figure 5-10.

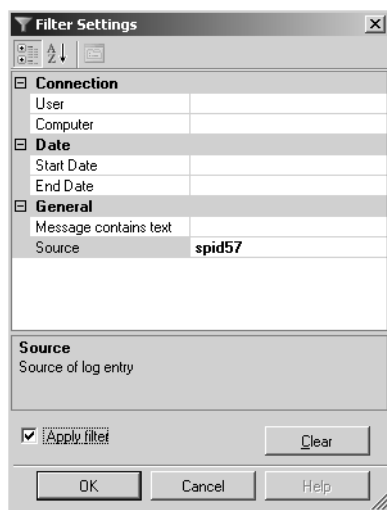


Figure 5-10 Specifying a filter.

6. Click OK. The filter is applied as shown in Figure 5-11. You should see any SPID57 events that are in the log. If no SPID57 events have been captured, the filtered log shows no events.

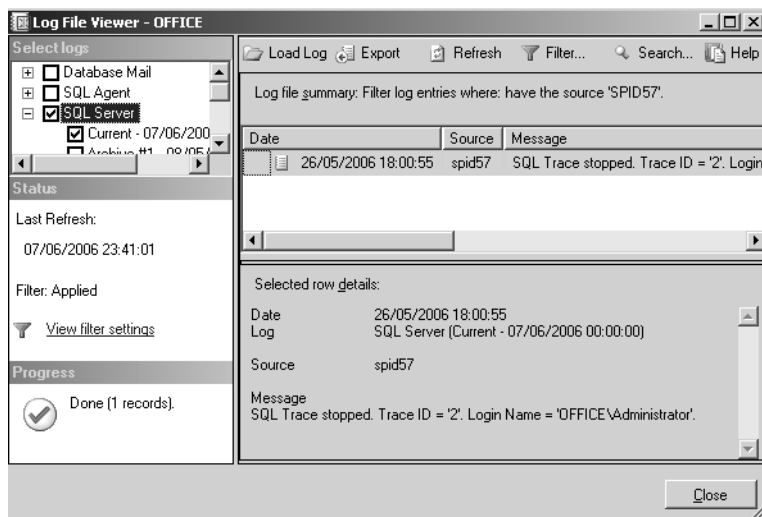


Figure 5-11 A filtered log.

► **Practice 2: Recycling the SQL Server Log**

In this practice, you start a new SQL Server log by using the *sp_cycle_errorlog* stored procedure.

1. Log in to your domain at your member server by using either your domain administrator account or an account that has been added to the sysadmin server role. If you are using virtual machine software, log in to your domain and connect to your member server.
2. From Programs, Microsoft SQL Server 2005, click SQL Server Management Studio. Connect to the database engine on your member server, specifying Windows Authentication. Connect using TCP/IP, and specify the AdventureWorks database.
3. Click New Query. In Query Editor, type **sp_cycle_errorlog** as shown in Figure 5-12.

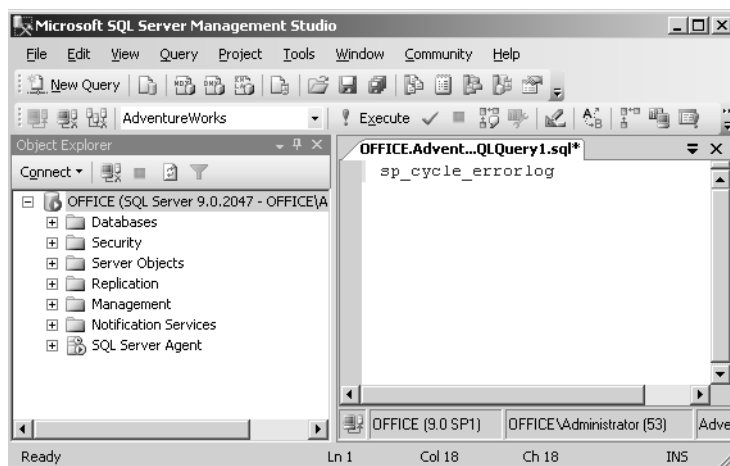


Figure 5-12 Specifying the stored procedure.

4. Click Execute (or press F5).
5. View the current SQL Server log by using the procedure specified in Practice 1. The reinitialized SQL Server log is shown in Figure 5-13.

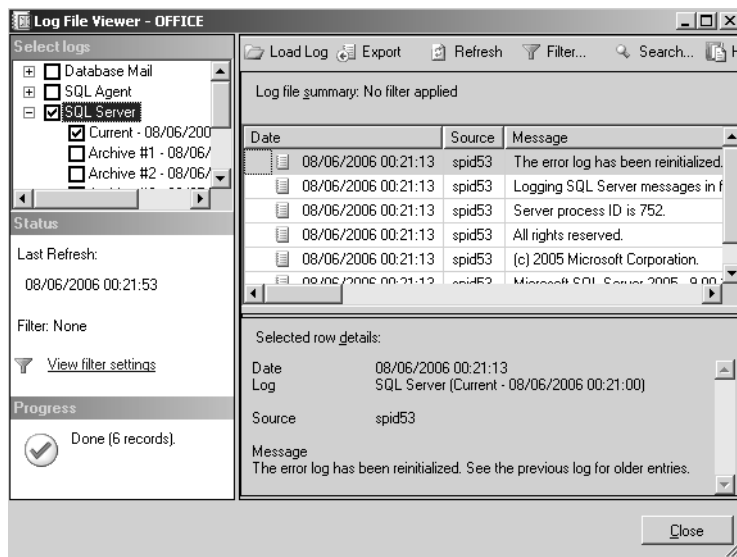


Figure 5-13 The reinitialized SQL Server log.

Lesson Summary

- The *sys.dm_exec_query_stats* server-scoped DMV returns aggregate performance statistics for cached query plans.
- You can view the SQL Server error log to ensure that processes have completed successfully (for example, backup and restore operations, batch commands, and scripts). The *sp_cycle_errorlog* system stored procedure can be used to cycle the error log files without having to restart the instance of SQL Server.
- The *sys.dm_os_wait_stats* DMV returns information about waits encountered by threads that are in execution. The *SQLServer:Wait Statistics* performance object contains performance counters that report information about wait status.
- A SQL trace contains event categories and describes an event class within a category by using data columns. SQL traces can be generated by using system stored procedures, but typically SQL Server Profiler is used for this purpose.
- Service Broker can be used to implement notifications that inform you when services have started.

- When you start an instance of the SQL Server database engine, you are starting the SQL Server service. After you start the SQL Server service, users can establish new connections to the server.
- The SQL Server Agent service runs jobs, monitors SQL Server, processes alerts, and allows you to automate administrative tasks. The service must be running before local or multi-server administrative jobs can run automatically.
- Failover clustering in SQL Server 2005 is implemented by the MSCS service. You can configure a SQL Server instance on one node of a failover cluster to fail over to any other node in the cluster during a hardware failure, operating system failure, or planned upgrade.

Lesson Review

You can use the following questions to test your knowledge of the information in Lesson 2, “Choosing the Appropriate Information to Monitor.” The questions are also available on the companion CD if you prefer to review them in electronic form.

NOTE Answers

Answers to these questions and explanations of why each answer choice is right or wrong are located in the “Answers” section at the end of the book.

1. You want to find out the last time a particular query was executed, the number of times the query has been executed, and the resources consumed by all invocations of the query plan, as well as the least and greatest amount of CPU consumed by a single invocation of the query plan. How do you do this?
 - A. Access the SQL Server log.
 - B. Access the Windows application log.
 - C. Use the *sys.dm_exec_query_stats* DMV.
 - D. Use the *sys.dm_os_wait_stats* DMV.
2. Which SQL Server database engine event category contains event classes that are produced when data or log files grow or shrink automatically?
 - A. Database
 - B. Objects
 - C. Full Text
 - D. Locks

3. You are monitoring the performance of a database application. You want to find the server process and transaction identities of transactions that are using excessive CPU resource. How should you do this?
 - A. Inspect the SQL Server log.
 - B. Inspect the SQL Server Agent log.
 - C. Use the *sys.dm_os_wait_stats* DMV.
 - D. Use Profiler to capture a SQL trace.
4. You need to monitor the MSCS service on a Microsoft failover cluster that is running SQL Server 2005. What logs should you inspect? (Choose all that apply).
 - A. The Windows security event log
 - B. The cluster log
 - C. The Windows application event log
 - D. The SQL Server log
 - E. The Windows system event log
 - F. The SQL Server Agent log

Chapter Review

To further practice and reinforce the skills you learned in this chapter, you can perform the following tasks:

- Review the chapter summary.
- Review the list of key terms introduced in this chapter.
- Complete the case scenarios. These scenarios set up real-world situations involving the topics of this chapter and ask you to create a solution.
- Complete the suggested practices.
- Take a practice test.

Chapter Summary

- Monitoring involves isolating problems and tracking performance trends. You need to take performance measurements at regular intervals over time to establish a server performance baseline and to track trends.
- You can define and create SQL traces by using SQL Server Profiler or Transact-SQL system stored procedures. A SQL trace contains event categories and describes an event class within a category by using data columns.
- You can use the Windows Performance Logs and Alerts tool to create an alert that triggers when a threshold performance counter value is reached. Event notifications execute in response to DDL statements and SQL Trace events and send information about these events to Service Broker.
- You can obtain aggregate performance statistics for cached query plans by using the *sys.dm_exec_query_stats* server-scoped DMV. You can obtain information about waits by using the *sys.dm_os_wait_stats* DMV and the *SQLServer:Wait Statistics* performance object.
- You can view the SQL Server error log to ensure that processes have completed successfully. Service Broker can be used to implement notifications that inform you when services have started. When you start an instance of the SQL Server database engine, you start the SQL Server service and users can establish new connections to the server.
- The SQL Server Agent service runs jobs, monitors SQL Server, processes alerts, and allows you to automate administrative tasks. Failover clustering in Microsoft SQL Server 2005 is implemented by the MSCS service.

Key Terms

Do you know what these key terms mean? You can check your answers by looking up the terms in the glossary at the end of the book.

- baseline
- event category
- event class
- metric
- monitoring
- Microsoft Cluster Server (MSCS) service
- notification
- Service Broker
- SQL Server log
- threshold value
- timestamp
- trace flag

Case Scenarios

In the following case scenarios, you apply what you've learned about defining and implementing monitoring standards for a physical server and choosing the appropriate information to monitor. You can find answers to these questions in the “Answers” section at the end of this book.

Case Scenario 1: Automating, Monitoring, and Configuring Alerts

You are a senior DBA at Litware, Inc. You are tasked with designing a monitoring strategy that generates baselines and monitors the health of the Litware SQL Server 2005 servers and the efficiency of applications and ad hoc queries that run against the tables in the Litware databases. You need to automate the monitoring process as much as possible. Your team members need to be warned if a transaction log is filling the space available to it, or if 75 percent or more of the capacity of a logical volume is being used. Litware uses hardware RAID10 volumes for both database and transaction log storage. You need to track trends and justify budgeting for future expenditure. You need to display the results of your monitoring in graphical form to non-technical management.

1. Transaction logs are not configured to grow automatically. You need to configure an alert that sends a message to your team members when any transaction log exceeds 70 percent of the space allocated. On what counter or counters should you set an alert?
2. You need to configure an alert to warn your team when the free space on the RAID10 array that holds the transaction logs for a particular database falls below 30 percent. You also need the alert to take action that would remedy the situation. What should you do?
3. How should you store the output from the counter log that you use to track trends?

Case Scenario 2: Identifying Slow and Resource-Intensive Transactions

You are the DBA at Trey Research. A team of application developers runs applications and ad hoc queries against tables and indexed views in Trey Research's databases. You need to identify which transactions take a long time to complete or use an excessive amount of server resources. Often you need to identify the developer who ran a particular transaction. You need to keep a record of when the SQL Server service and the SQL Server Agent service starts and stops, and when the EXECUTE AS statement is used. You also need to record the batch commands and scripts that complete successfully.

1. What is the best way for you to identify that the transactions take a long time to complete or use an excessive amount of server resources, and who the developer is that is running these transactions?
2. You are configuring notifications that inform your team when the SQL Server service and the SQL Server Agent service starts and stops, and when the EXECUTE AS statement is used. What event classes do you use to trigger these notifications?
3. What is the easiest way of determining which batch commands and scripts complete successfully?

Suggested Practices

To help you successfully master the exam objectives presented in this chapter, complete the following tasks.

Define and Implement Monitoring Standards for a Physical Server

You should complete these practices on your test network before implementing monitoring on your production servers. The results you get will vary because your test network is not handling the same traffic patterns as your production servers.

- **Practice 1** Configure a Performance counter log to capture selected counters from those listed in Lesson 1 of this chapter. Capture a baseline when the test network is not running any queries, and capture the result in .cdf format. Capture the log again when you are running queries against the AdventureWorks database. If you have any stress-testing software, you can use it to stress your network. Use a .cdf file reader (for example, Microsoft Excel) to compare your results.
- **Practice 2** Configure a Performance counter log to capture the performance objects listed in Lesson 1 of this chapter, and configure the log to write its results to a database. Capture a baseline when the test network is not running any queries. Capture the log again when you are running queries against the AdventureWorks database. If you have any stress-testing software, you can use it to stress your network. Use SQL Server queries to compare your results.
- **Practice 3** Configure a notification to warn you when the SQL Server Agent service stops, starts, or pauses. Test the notification by stopping and starting the service.

Choose the Information to Monitor

You should complete these practices on your test network before implementing monitoring on your production servers. If your test hardware cannot support an MSCS active/passive cluster, do not attempt the final practice.

- **Practice 1** Run a query against the AdventureWorks database, and use the *sys.dm_exec_query_stats* server-scoped DMV to obtain aggregate performance statistics for the query plan.
- **Practice 2** Run a number of queries against the AdventureWorks database, or use stress-testing software. Use the *sys.dm_os_wait_stats* DMV to obtain information about any waits that occurred. Check the SQL Server and SQL Server Agent error logs.
- **Practice 3** Configure your test hardware to implement an active/passive MSCS cluster running SQL Server 2005. Stop the active node, and fail over to the second node. Restart the first node, and allow failback to occur. Check the cluster log, and the Windows application and system logs.

Take a Practice Test

The practice tests on this book's companion CD offer many options. For example, you can test yourself on just one exam objective, or you can test yourself on all the 70-444 certification exam content. You can set up the test so that it closely simulates the experience of taking a certification exam, or you can set it up in study mode so that you can look at the correct answers and explanations after you answer each question.

MORE INFO Practice tests

For details about all the practice test options available, see the "How to Use the Practice Tests" section in this book's Introduction.

Chapter 6

Database Maintenance

Database maintenance is an all-encompassing term for what database administrators (DBAs) spend most of their time doing while they are at work. As with a car, the performance of a database that does not receive regular maintenance degrades over time. Just as not checking tire pressure and changing oil filters can influence automobile performance, failure to defragment indexes, reclaim free space, and update statistics cause the database to become less responsive. This chapter looks at the methods and technologies a DBA can use to keep her database functioning at peak performance. Readers learn how Microsoft SQL Server assists with automating many maintenance tasks and how they can take things further by developing more individualized database maintenance plans.

Exam objectives in this chapter:

- Create and implement a maintenance strategy for database servers.
 - Create a job dependency diagram.
 - Manage the maintenance of database servers.
- Design a database maintenance plan.
- Design a strategy to manage Reporting Services.
- Design a strategy to manage data across linked servers.
- Set up and manage linked servers.

Lessons in this chapter:

- Lesson 1: Creating and Implementing a Maintenance Strategy for Database Servers 305
- Lesson 2: Designing a Database Maintenance Plan 317
- Lesson 3: Managing Reporting Services 331
- Lesson 4: Designing a Strategy to Manage Data Across Linked Servers 351

Before You Begin

To complete the lessons in this chapter, you must have completed the following tasks:

- Configured a Microsoft Windows Server 2003 R2 computer with Microsoft SQL Server 2005 Enterprise Edition SP1 as detailed in the Appendix
- Installed an updated copy of the AdventureWorks sample database as detailed in the Appendix

No additional configuration is required for this chapter.

Lesson 1: Creating and Implementing a Maintenance Strategy for Database Servers

Most mechanics don't pick up a tool and start looking for things to fix on a car. The best mechanics plan their approach carefully, knowing how they will approach maintenance tasks before they pick up any tools. Without planning maintenance, a mechanic might end up taking apart an entire engine to discover that it is working perfectly. This lesson concentrates on the steps that you can take to plan maintenance operations, including understanding a database's structure and planning maintenance tasks. It also includes preventative maintenance, that is, setting up strategies within the database to stop problems before they happen.

After this lesson, you will be able to:

- Prepare DDL triggers to protect vital aspects of a database.
- Create database diagrams to understand the interrelationships between database objects.
- Create job dependency diagrams to help in the planning of maintenance tasks.
- Prepare the operating system for the application of hotfixes, updates, and service packs.
- Prepare the database software for the application of hotfixes, updates, and service packs.

Estimated lesson time: 30 minutes

Capturing Data Definition Language (DDL) Operations Using DDL Triggers

You can use data definition language (DDL) triggers as a preventative maintenance technology. DDL triggers execute stored procedures in response to Transact-SQL statements that start with a particular set of keywords that are most often—but not always—CREATE, ALTER, and DROP. From the perspective of database maintenance, you can configure DDL triggers to perform the following tasks:

- Prevent changes to database schema.
- Record changes or events in the database schema.
- Perform another action in response to a change in database schema.

DDL triggers can fire only after the DDL statements that initiate them are executed. You cannot configure a DDL trigger as an INSTEAD OF trigger, though DDL triggers

can roll back database changes such as the dropping of a table. The other sort of trigger—DML triggers, which include INSTEAD OF and AFTER triggers—are covered more completely in Chapter 8, “Design Data Integrity.”

MORE INFO DDL events

A full list of DDL events that can be used to fire DDL triggers can be found in the following MSDN article: [msdn2.microsoft.com/en-us/library/ms189871\(d=ide\).aspx](https://msdn2.microsoft.com/en-us/library/ms189871(d=ide).aspx).

The following statement is an example of how you can use a DDL trigger to protect a database’s tables from being altered or dropped:

```
CREATE TRIGGER Protect
ON DATABASE
FOR DROP_TABLE, ALTER_TABLE
AS
    PRINT 'Warning: Before dropping or altering tables, disable trigger "Protect"!'
    ROLLBACK ;
```

Real World

Orin Thomas

Although correctly set roles and permissions are a good guard against users breaking mission-critical tables, as an administrator you should learn to be paranoid. Configure DDL triggers to prevent “user events” from occurring that you would otherwise have to spend hours fixing yourself.

You can configure DDL triggers to fire in response to Transact-SQL events applied against the current database or the current server. The scope of the trigger is dependent on the event that fires it. The DDL trigger in the previous example is limited to the database that it is applied to. The next example is a DDL trigger that has the entire server instance as its scope. It will print a message if a CREATE LOGIN, ALTER LOGIN, or DROP LOGIN statement is issued against the current instance.

```
CREATE TRIGGER login_trigger
ON ALL SERVER
FOR DDL_LOGIN_EVENTS
AS
    PRINT 'A LOGIN STATEMENT HAS BEEN ISSUED.'
    SELECT EVENTDATA().value('(/EVENT_INSTANCE/TSQLCommand/CommandText)[1] ',
'nvarchar(max)');
```

DDL triggers that have the database as a scope are stored as objects within the database against which they apply. You can find information about DDL triggers using the

`sys.triggers` catalog view from within the database context in which they are created. DDL triggers that have the server instance as the scope are stored as objects within the master database. You can obtain information about server-scoped triggers using the `sys.server_triggers` catalog view in any database context.

You can also configure DDL triggers to fire after the execution of any Transact-SQL event that belongs to a predefined grouping of similar events. For example, rather than having to specify that a DDL trigger should fire after a `CREATE TABLE`, `ALTER TABLE`, or `DROP table` statement, you can use `FOR DDL_TABLE_EVENTS` in the `CREATE TRIGGER` statement to encompass all three.

MORE INFO Event groups

A full list of event groups and the statements they encompass can be found in the following MSDN article: [msdn2.microsoft.com/en-us/library/ms191441\(d=ide\).aspx](https://msdn2.microsoft.com/en-us/library/ms191441(d=ide).aspx).

Creating Database Diagrams

Database diagrams allow a DBA to gain an understanding of the complex interrelationships between database tables. Although you can gain this understanding by examining the properties of individual tables, a visual representation, such as the partial AdventureWorks database diagram shown in Figure 6-1, simplifies the process of checking how database objects interrelate. Database administrators who understand these interrelationships are better equipped to handle maintenance tasks because they can better ascertain how critical certain database objects are to the database as a whole. This knowledge informs the DBA which objects should be prioritized during maintenance.

To create a database diagram, perform the following tasks:

1. Open SQL Server Management Studio, and connect to the appropriate server instance.
2. Expand the Databases folder.
3. Expand the folder of the database that you want to diagram.
4. Select the Database Diagrams folder. If this is the first time you have tried to create a database diagram, a message appears stating that the database does not have one or more of the support objects required to use database diagramming and asking whether you want to create them. Click Yes.
5. Right-click the Database Diagrams folder and choose New Database Diagram.
6. A list of tables within the database is shown in the Add Table dialog box. Use the scroll bar to scroll to the bottom, and then press Shift and click the final entry to select all tables. Click Add.

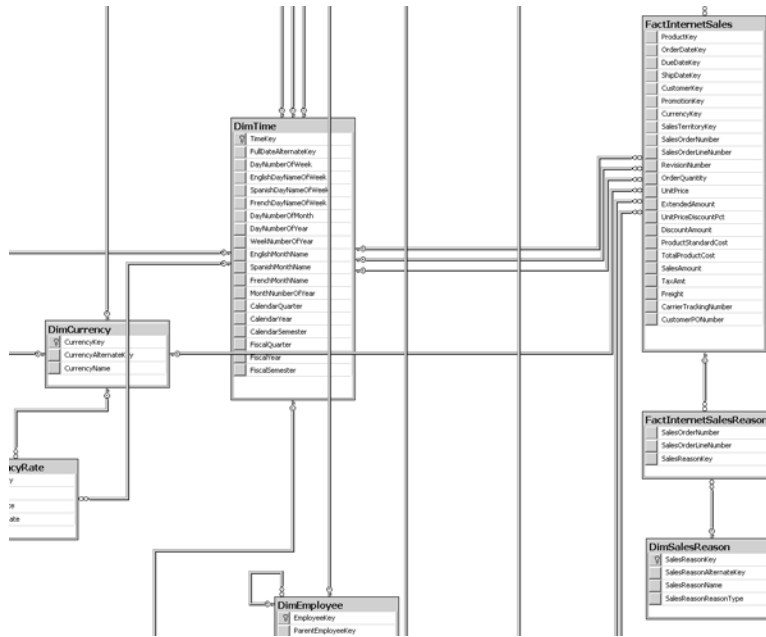


Figure 6-1 Part of the AdventureWorks database diagram.

7. SQL Server Management Studio (SSMS) generates the database diagram, including all selected tables. Click Close to close the Add Table dialog box.
8. Use the View menu in SQL Server Management Studio to zoom in and out to view the entire diagram or individual details.
9. To save the diagram, choose Save from the File menu of SQL Server Management Studio.

You can also create diagrams that focus on specific tables and their interrelationships. It is usually better to do this after you have created a whole database diagram so that you can better determine where to focus. The practice at the end of the lesson covers this task in more detail.

Job Dependency Diagrams

Job dependency diagrams are a visual way of representing complicated flow information. Diagrams can be more effective than using lists because presenting branching options in a graphical form is easier than presenting them in a linear format. This understanding drives the interface of the Business Intelligence Development Studio (BIDS), which is covered in more detail in Chapter 7.

Job dependency diagrams include information about timelines and dependencies. This information makes it easy to determine the order in which tasks execute, as well as which tasks should execute based on the results of previously executed tasks.

Consider, for example, a typical sequence of events at a research organization that performs many scientific analyses each week. The job dependencies can be described as follows. Once the results of the scientific analyses are forwarded to clients, the data should be archived by copying it to a special archive table because it is unlikely to be needed again. When it is archived, the data should be removed from the analysis table. A consistency check should then be performed on the database, and an operator should be notified of the result. If the data archiving fails, this failure should be reported, but the consistency check should still be performed. Similarly, if the analysis data is not deleted, the consistency check should still be performed. Finally, the results of the database consistency check should be reported under all circumstances.

Although this example is reasonably straightforward, you might need to read it several times because text is a linear medium that isn't ideal for expressing branching information. As the nature of the job that is being planned gets more complex, the text used to describe it becomes more convoluted. Luckily, you can represent complex information more simply by using a job dependency diagram like the one shown in Figure 6-2. In job dependency diagrams, items are linked either by success, failure, or completion. These same links are used for workflow tasks when creating a custom maintenance plan or a SQL Server Integration Services (SSIS) package in BIDS.

When creating a job dependency diagram, remember to separate each task and then determine links to later tasks based on the success and failure of the previous task. In some situations, the workflow should not terminate when a task fails. In the previous example, the database consistency check should occur even if earlier tasks had failed. Although there is no one correct diagram for a set of tasks, after you have completed a job dependency diagram, the diagram is likely to be good if implementing the job is easier than it would be with only a list of instructions.

MORE INFO SQL Server Profiler

SQL Server Profiler is an excellent tool for monitoring SQL Server 2005. Complete coverage of this tool, including how to monitor and resolve deadlocks, is provided in Lessons 2 and 3 in Chapter 1, "Troubleshooting Database and Server Performance." Administrators looking for more information should consult the following MSDN article: [msdn2.microsoft.com/en-us/library/ms187929\(d=ide\).aspx](http://msdn2.microsoft.com/en-us/library/ms187929(d=ide).aspx).

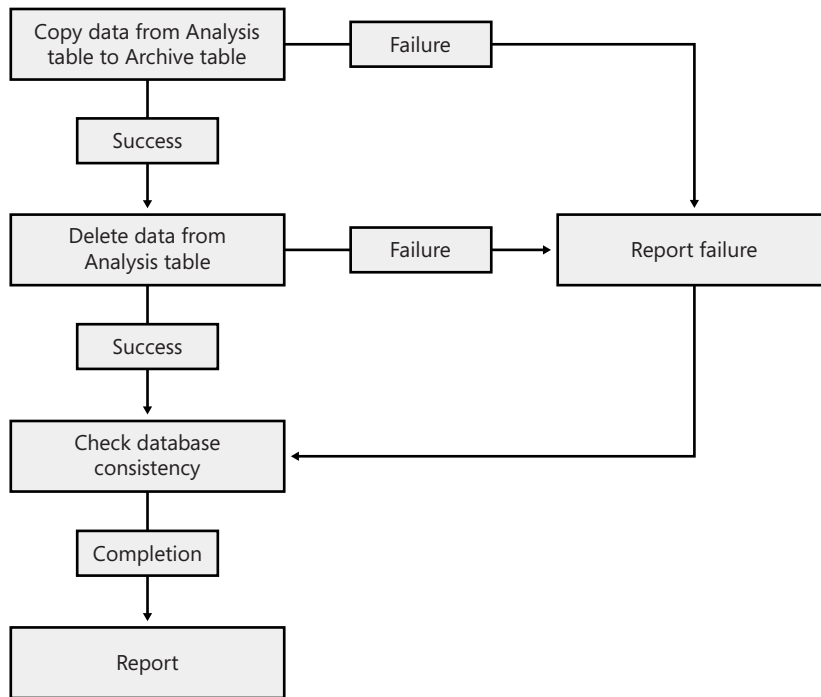


Figure 6-2 A basic job dependency diagram.

Quick Check

1. When do DDL triggers fire?
2. Where are DDL triggers stored if their scope is the instance?

Quick Check Answer

1. DDL triggers fire after the triggering statement has been executed.
2. DDL triggers whose scope is the instance are stored in the master database.

Applying Service Packs, Software Updates, and Security Updates

Applying updates—be they service packs, software updates, or security updates—is a significant part of the maintenance cycle. All administrators should subscribe to the Microsoft Security Bulletins at www.microsoft.com/technet/security/bulletin/notify.msp. These e-mail notifications provide information about newly released bulletins that will be relevant to the platforms that database administrators are responsible for managing. The remainder of this section describes the processes that administrators should go through prior to deploying updates to production database servers.

Update Management Infrastructure

Although you can download updates from the Microsoft Web site on an as-needed basis, you'll likely find that deploying a single server on a network's screened subnet running Microsoft Windows Server Update Services (WSUS) is more efficient with respect to bandwidth. A WSUS server located on a screened subnet provides a centralized location for storing updates, security hotfixes, and service packs. WSUS Service Pack 1 not only supports updates for Windows operating systems such as Windows Server 2003, but it also supports updates for SQL Server 2005. Using WSUS means that updates need to be downloaded only once from Microsoft's Internet servers. Clients on an organization's network can then download updates from the WSUS server. Because some updates and service packs can be several hundred megabytes in size, downloading the update only once from Microsoft's Internet servers, rather than once for each computer on the network, can significantly decrease the amount of bandwidth used by an organization. WSUS also allows an administrator to choose which updates are approved for deployment across an organization. This flexibility allows a DBA to extensively test updates before they are deployed to production computers.

MORE INFO WSUS

To find out more information about WSUS, consult the following Web site: www.microsoft.com/windows-serversystem/updateservices/evaluation/SP1overview.aspx.

The Microsoft Baseline Security Analyzer (MBSA) tool, which can be used in conjunction with WSUS, allows administrators to scan computers to check whether they have the latest updates applied. The current version of the MBSA tool supports checking both Windows Server 2003 and SQL Server 2000 to determine which currently released updates have yet to be applied. At present, SQL Server 2005 has only basic support, but full support is likely to be present in the next version of the tool. Figure 6-3 shows the results of an MBSA scan against a Windows Server 2003 computer running SQL Server 2005.

MORE INFO Microsoft Baseline Security Analyzer

For more information about the Microsoft Baseline Security Analyzer, consult the following Web site: www.microsoft.com/technet/security/tools/mbsahome.aspx.

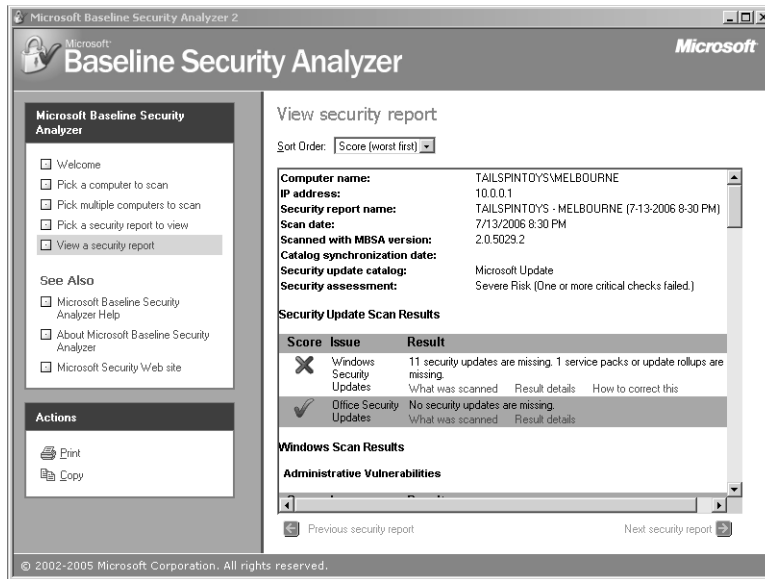


Figure 6-3 The results of an MBSA scan.

It is also in a database administrator's interest to ensure that automatic updates are disabled on computers running SQL Server 2005. The reason for this is simple: on important production systems, you should test all updates thoroughly before they are applied. If automatic updates are configured to automatically install, it is possible that an untested update will be applied to a system. This can lead to highly unexpected results.

Update Testing

SQL Server 2005 databases are often critical to an organization's operation. Database administrators should remember this fact and repeat it when they have to approach management about setting up a update-testing infrastructure. Prior to applying any security update, hotfix, or service pack, a DBA should perform rigorous testing. By performing such testing, a DBA can ensure that an update that is supposed to fix one problem does not create a whole host of others. The best way to test any security update, hotfix, or service pack is on a configuration identical to the one that is in production. Personnel in the accounting department might oppose the idea of buying two expensive database servers instead of one, but the DBA needs to emphasize that this hardware is critical to the organization's operation.

In the real world, this logical argument doesn't always succeed. Administrators are often reduced to testing hotfixes, security updates, and service packs on spare workstations or, more increasingly, on virtual servers that simulate the production

environment. When performing update testing prior to deployment, try to do the following:

- Prior to beginning the test, ensure that the same hotfixes, service packs, and security updates that are installed on the production computer's operating system are also installed on the test computer's operating system.
- Prior to beginning the test, ensure that the same hotfixes, service packs, and security updates that are installed on the production computer's operating system are also installed on the test computer's SQL Server 2005 instance.
- Try to have the hardware configuration on the test computer match the production computer's operating system as closely as possible.
- Try to ensure that the data stored on the test computer matches the data on the production computer as closely as possible. One way to do this is to restore the most recent full backup to the test computer.
- Try to have a small group of experienced database users interact with the test computer once the update is applied to determine whether any aspect of the system is behaving abnormally.

Applying the Updates

After you thoroughly test the update and find that it does not cause any problems, you can apply it to the production database. Prior to applying hotfixes, security updates, or service packs, you should perform the following steps:

- Perform a full operating system backup, including a backup of system state data using the Windows backup utility.
- Perform a full Automated System Recovery backup using the Windows backup utility.
- Perform a full backup of all databases, including the system databases, using SQL Server Backup.

A DBA should choose to apply the update at a time when having the database go offline is minimally disruptive to database users. Typically, one of the best times to perform this type of maintenance is at 2 A.M. on a Saturday because if something goes wrong, the DBA has the remainder of the weekend to rectify it. If something does go wrong, keep the following points in mind:

- Don't panic.
- Try to roll back the hotfix, update, or service pack, and see if that helps.

- Perform an Automated System Recovery.
- If worst comes to worst, the full backups that were taken enable you to rebuild the database from scratch.

PRACTICE Creating a Database Diagram for the AdventureWorks DW Database

To create a database diagram, perform the following tasks:

1. Open SQL Server Management Studio, and connect to the instance on the server Melbourne.
2. Expand the Databases folder.
3. Expand the folder of the AdventureWorks DW Database.
4. Select the Database Diagrams folder. If this is the first time you have tried to create a database diagram, a message appears stating that the database does not have one or more of the support objects required to use database diagramming and asking whether you want to create them. Click Yes. If you have already created a database diagram, proceed to step 5.
5. Right-click the Database Diagrams folder and choose New Database Diagram.
6. A list of tables within the database will be shown in the Add Table dialog box. Hold down the Ctrl key and select the DimEmployee, DimSalesTerritory, DimTime, and FactInternetSales tables. Click Add.
7. Click Close to dismiss the Add Table dialog box.
8. Navigate the new database diagram by using the scrollbars.

Lesson Summary

- DDL triggers can be configured to protect the database from user damage.
- DDL triggers can have a database or instance scope. The scope of a DDL trigger is dependent on the statement that fires the trigger.
- DDL triggers that have the scope of a single database are stored in that database. DDL triggers that have the scope of an entire instance are stored in the master database.
- Database diagrams provide visual representations of the interrelationships between database objects.
- Database diagrams can be created using SQL Server Management Studio.

- Job dependency diagrams provide an easy-to-use way of planning all the necessary steps involved in database maintenance tasks.
- Windows Server Update Services can be used to reduce the amount of bandwidth used in downloading updates from the Microsoft update servers.
- The MBSA tool can be used to determine which hotfixes and updates have yet to be applied to Windows Server 2003 computers.
- All hotfixes, software updates, and service packs should be rigorously tested on configurations similar to that of production computer's operating system prior to being applied to the production computer.
- Prior to installing hotfixes, software updates, and service packs, perform a full operating system backup, a full Automated System Recovery backup, and a full backup of all databases, including system databases.
- If something goes wrong, attempt to roll back the update first.

Lesson Review

You can use the following questions to test your knowledge of the information in Lesson 1, "Creating and Implementing a Maintenance Strategy for Database Servers." The questions are also available on the companion CD if you prefer to review them in electronic form.

NOTE Answers

Answers to these questions and explanations of why each answer choice is correct or incorrect are located in the "Answers" section at the end of the book.

1. Which of the following catalog views would you access to learn about DDL triggers that apply to all databases on a particular instance?
 - A. *sys.triggers*
 - B. *sys.server_triggers*
 - C. *sys.database_permissions*
 - D. *sys.database_principals*
2. If a DDL trigger that applies to a database uses FOR DDL_TABLE_EVENTS, which of the following Transact-SQL statements will cause it to fire? (Choose all that apply.)
 - A. CREATE TABLE

- B. ALTER TABLE
 - C. CREATE DATABASE
 - D. ALTER DATABASE
3. If problems occur after the installation of a service pack, which of the following steps should a database administrator take first?
- A. Roll back the service pack.
 - B. Perform Automated System Recovery.
 - C. Restore all system databases.
 - D. Restore all user databases.
4. Which of the following tools can be used to determine whether a Windows Server 2003 computer has the most recent service packs, hotfixes, and updates installed?
- A. SQL Server Management Studio (SSMS)
 - B. Windows Server Update Services (WSUS)
 - C. Microsoft Baseline Security Analyzer (MBSA)
 - D. Business Intelligence Development Studio (BIDS)

Lesson 2: Designing a Database Maintenance Plan

This lesson concentrates on the technologies that SQL Server 2005 provides to assist in creating and implementing maintenance plans. It covers the tasks that can be included in a maintenance plan generated by the Maintenance Plan Wizard. It introduces the building of custom maintenance plans for more experienced administrators. The lesson also covers the Transact-SQL statements required to create backup devices and to take full, differential, and transaction log backups.

After this lesson, you will be able to:

- Create a maintenance plan using the Maintenance Plan Wizard.
- Understand the tasks that can be accomplished using maintenance plans.
- Create a custom maintenance plan.
- Create file, network, and tape backup devices.
- Use Transact-SQL statements to perform full, differential, and transaction log backups.

Estimated lesson time: 30 minutes

IMPORTANT Related lesson material

Many topics mentioned in this lesson—such as pages, extents, and backup types—are more fully described in Lesson 2 of Chapter 4, “Disaster Recovery.” You should review that lesson fully before proceeding with this one.

Database Maintenance Plans

Database maintenance plans describe the workflow of maintenance tasks that ensure that a database performs well, gets backed up regularly, and is checked for inconsistencies. Maintenance plans create and configure jobs that perform maintenance tasks on the database at scheduled intervals. There are two methods of creating maintenance plans:

- **Use the Maintenance Plan Wizard.** This wizard allows for the creation of basic maintenance plans with little flexibility.
- **Create a Maintenance Plan Manually.** This approach allows for greater flexibility in the creation of plans. This method is recommended for more experienced administrators.

SQL Server 2005 does not support the creation of a single maintenance plan for multiple servers. You should create separate maintenance plans on each server.

Maintenance Plan Tasks

When you create a maintenance plan using the Maintenance Plan Wizard, you can include a pre-set list of tasks. Following is a list of these tasks and a description of configuration options they contain:

- **Check Database Integrity** This task performs internal consistency checks of data and index pages within the database. If this task is selected, the administrator can select all databases, all system databases, all user databases, or a combination of these options.
- **Shrink Database** This task reduces the disk space consumed by the database and log files by removing empty data and log pages. If this task is selected, the administrator can select all databases, all system databases, all user databases, or a combination of these options. The administrator can select a size threshold for when the Shrink Database task will execute and for the amount of free space remaining after the shrink. Free space can be retained in the database files or released to the operating system.
- **Reorganize Index** This task defragments and compacts clustered and nonclustered indexes on tables and views. Doing so improves index-scanning performance. If this task is selected, the administrator can select indexes in all databases, all system databases, all user databases, or a combination of these options.
- **Rebuild Index** This task reorganizes data on the data and index pages by rebuilding indexes to improve performance. Indexes from all, none, or a combination of databases can be rebuilt. As shown in Figure 6-4, you can reorganize pages with the default amount of free space or specify a free-space-per-page percentage. Advanced options include the ability to pad an index, sort results in tempdb, ignore duplicate keys, and keep an index online during the reindexing process.
- **Update Statistics** This task ensures that the query optimizer has up-to-date information about the distribution of data values within tables, improving performance. Statistics from all, none, or a combination of databases can be

updated. You can configure this task to update all existing statistics, column statistics only, or index statistics only.

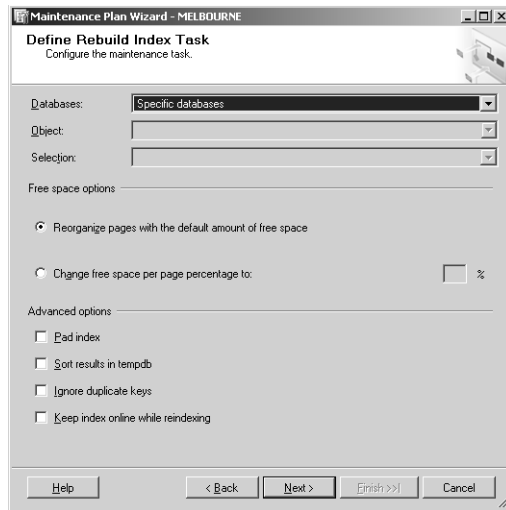


Figure 6-4 Configuring the settings for the Rebuild Index maintenance task.

- **Clean Up History** This task deletes historical data about SQL Server Agent jobs, Maintenance Plan operations, and Backup and Restore. As shown in Figure 6-5, you can configure the age after which historical data will be deleted.

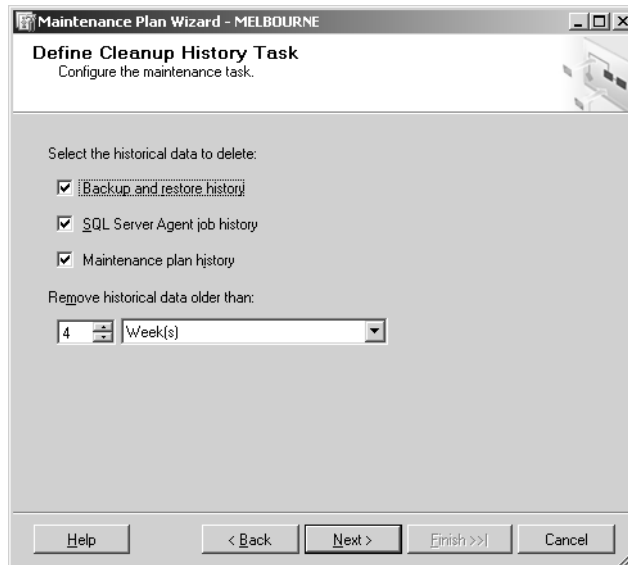


Figure 6-5 Selecting which historical data should be periodically deleted.

- **Execute SQL Server Agent Job** This task allows for SQL Server Agent jobs, which include SSIS packages, to be run as part of the maintenance plan. Only existing SQL Server Agent jobs can be run as a part of this maintenance task.
- **Back Up Database (Full)** This task allows a full database backup on the specified database. You must specify the backup destination and overwrite options.
- **Back Up Database (Differential)** This task allows a differential backup of database extents that have been modified since the last full database backup.
- **Back Up Database (Transaction Log)** This task performs a backup of the database's transaction log, which is necessary if a database is configured to use the full or bulk-logged recovery model.

Creating a Maintenance Plan Using the Maintenance Plan Wizard

To use the Maintenance Plan Wizard to create a maintenance plan, perform the following steps:

1. Open SQL Server Management Studio, and connect to the appropriate server instance.
2. If necessary, expand the server instance.
3. Expand the Management folder.
4. Right-click the Maintenance Plans folder and choose Maintenance Plan Wizard. The Maintenance Plan Wizard starts as shown in Figure 6-6.

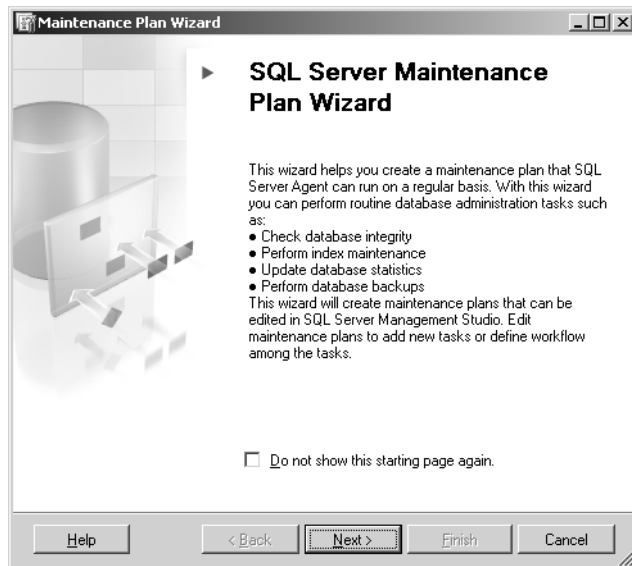


Figure 6-6 Maintenance Plan Wizard.

5. Click Next.
6. Enter a name for the maintenance plan and a description. Ensure that the Server field is set to the instance on which you want the maintenance plan to run. If the server is configured to use Windows Authentication, ensure that SQL Server Authentication is not selected. If the instance is configured to use SQL Server Authentication, enter the appropriate credentials. Click Next.
7. On the Select Maintenance Tasks page, select the check boxes for the tasks that should be performed as a part of this maintenance plan. In general, you should select only a few tasks for each maintenance plan. The more convoluted a single maintenance plan is, the more likely it becomes that an aspect of it will fail. The failure of one component in a complex maintenance plan means that many components might not execute. If maintenance plans with fewer components are used, the failure of one component will not have as drastic an impact. Click Next.
8. The Select Maintenance Task Order page, shown in Figure 6-7, requires you to determine the order in which tasks are executed. If necessary, select tasks, and click the Move Up or Move Down buttons to reorder them. Click Next.

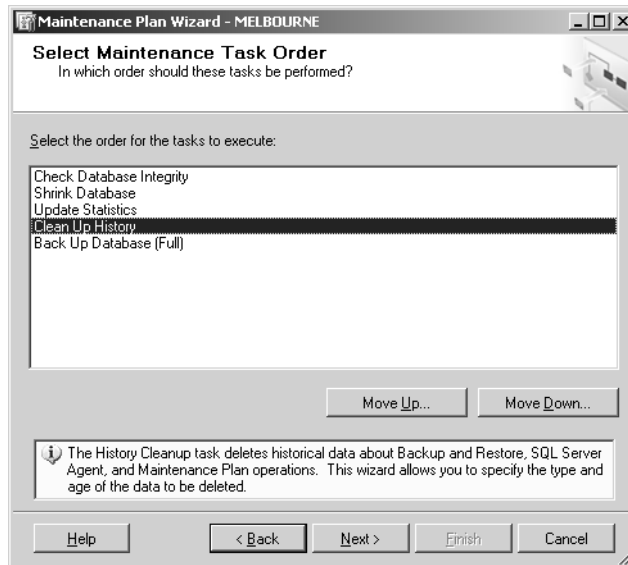


Figure 6-7 The Select Maintenance Task Order page.

9. The pages that appear next depend on which maintenance tasks you selected in the previous steps. You must configure each maintenance task appropriately. For example, if you select the Check Database Integrity task, you next see the

Define Database Check Integrity Task page. On this page, you must select which databases on the instances you want to check. Your options include All Databases, All System Databases, All User Databases, and specifically selected databases.

10. After you have configured the tasks, you need to specify a schedule on the New Job Schedule dialog box. The default schedule is to allow the plan to run as needed. If you specify a schedule, you must at least enter a schedule name before clicking OK.
11. Click Next.
12. The Select Report Options page enables you to have a report on the maintenance plan either written to a text file or sent as an e-mail to a particular recipient. After you have configured these options, click Next.
13. A summary page is displayed explaining what actions will occur when the maintenance plan tasks run. Click Finish to close the summary page.
14. The maintenance plan is then created, as shown in Figure 6-8. Click Close to end the Maintenance Plan Wizard.

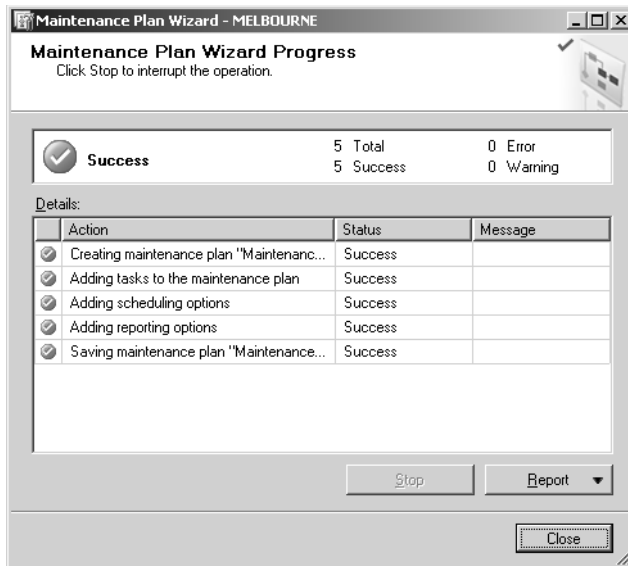


Figure 6-8 The final screen confirms that the maintenance plan has been created.

15. The newly created maintenance plan should be visible under the Maintenance Plans folder in SQL Server Management Studio.

Creating a Maintenance Plan Without the Wizard

Creating a maintenance plan without a wizard is similar to creating a package in BIDS, which is covered in Chapter 7, “SQL Server Integration Services.” Maintenance plans are built by dragging task flow elements from a toolbox onto the maintenance plan design surface. The maintenance plan tasks toolbox is shown in Figure 6-9.

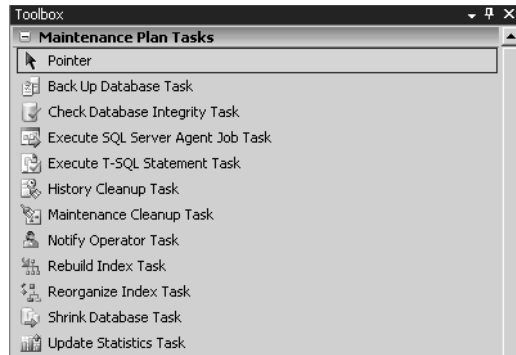


Figure 6-9 The maintenance plan tasks toolbox.

Once a task is dragged to the maintenance plan design surface, you need to double-click the task to configure it. You can view the Transact-SQL code that will be used in each task’s execution, which is an option not available with Maintenance Plan Wizard items. However, only experienced database administrators are likely to be interested in such details.

Creating a maintenance plan using this interface allows an administrator to configure conditional task branching that is dependent on the execution status of a previous task. For example, an administrator might want to perform a database integrity check. The DBA might want to be notified by the maintenance plan if that check fails so that she can terminate the execution of the task. If the database integrity check passes, the administrator might want the next task in the plan to execute without providing notification.

You connect maintenance tasks by dragging arrows from one task to another. Once the tasks are connected, right-click the line between the two tasks and select Success (the default), Failure, or Completion, as shown in Figure 6-10. If you select Failure, the line connecting the two tasks becomes red. If you select Completion, the line connecting the two tasks becomes blue. There can be multiple connections so that more than one task can be executed when a previous task succeeds, fails, or completes.

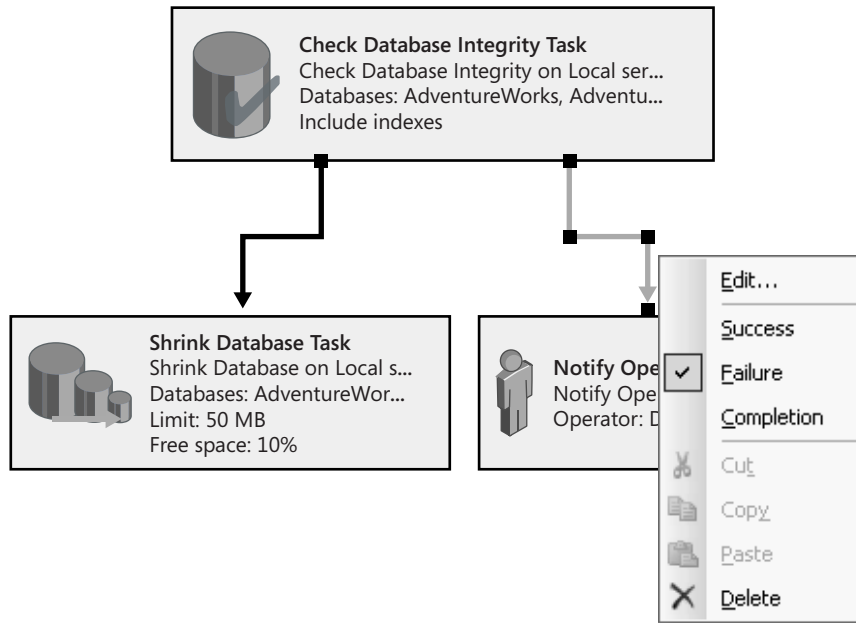


Figure 6-10 A maintenance task configured to notify the operator when a Check Database Integrity task fails and to execute the Shrink Database task if the Check Database Integrity task succeeds.

Tasks created using the Maintenance Plan Wizard can be executed only in a linear fashion. Creating tasks without using the Maintenance Plan Wizard allows for more complex task execution logic and a greater ability to customize tasks.

Quick Check

1. Which maintenance plan task can be used to defragment clustered indexes?
2. Which maintenance plan task frees up space?

Quick Check Answer

1. The Reorganize Index task can be used to defragment clustered indexes.
2. The Shrink Database task can be used to free up space.

Database Backups

Although creating a maintenance plan to perform backup operations is relatively simple, DBAs should be aware of the Transact-SQL statements that can perform these operations. In some cases, you will need to interface with the database by using Transact-SQL rather than using a GUI wizard.

Creating a Backup Device

Although you can perform a backup by explicitly specifying a location, performing regular backups by having them written to a preconfigured backup device is easier. These devices act like aliases for SQL Server. The *sp_addumpdevice* stored procedure adds a backup device to the *sys.backup_devices* catalog view. Once you add the device, you can then directly refer to it in BACKUP and RESTORE Transact-SQL statements. After you create the logical device, you do not need to use the “TAPE=” or “DISK=” clauses to specify a device path.

CAUTION Ownership permissions

Incorrectly specified ownership and permissions will interfere with the correct operation of disk or file backup devices. Administrators should ensure that appropriate permissions are applied to the Windows account under which the Database Engine was started.

The following three examples show how you can use the *sp_addumpdevice* stored procedure to create a disk backup device, tape backup device, and network share backup device.

Adding a Disk Backup Device To create a disk backup device named *diskback* that will back up files to *c:\backups\diskback.bak*, have the master database in focus and execute the following Transact-SQL statement:

```
EXEC sp_addumpdevice 'disk', 'diskback', 'c:\backups\diskback.bak';
```

If a volume fills during a backup operation, the backup operation will fail. You can also write disk backups to sets of hot-swappable disks that allow them to transparently replace backup disks when a backup is taken.

BEST PRACTICES Be careful where you write your backups.

If you are configuring SQL Server 2005 to back up to disk, ensure that it is not the same disk that transaction logs, database files, or program files are installed on. If you store backups on the same disk as the database files and the disk fails, you will lose both your database files and the backups of those database files!

Adding a Tape Backup Device To create a tape backup device named *tapeback* to the device that has the physical name *\\.\tape0*, have the master database in focus and execute the following Transact-SQL statement:

```
EXEC sp_addumpdevice 'tape', 'tapeback', '\\.\tape0';
```

To view a list of tape devices installed on the computer that hosts the SQL Server 2005 instance, query the `sys.dm_io_backup_tapes` dynamic management view. You cannot back up to a tape drive connected to another computer, even if that computer is running SQL Server 2005. If a tape backup device is filled during a backup operation and there is still more data to be written, SQL Server prompts for a new tape and then continues the backup operation when the new tape is loaded.

Adding a Network Share Backup Device To create a network share as a backup device, the network share and NTFS permission must be configured with write access for the user account under which the database engine was started. To create a network share backup called *networkback* for the `\\<servername>\<sharename>\filename.bak`, have the master database in focus and execute the following Transact-SQL command:

```
EXEC sp_addumpdevice 'disk', 'networkback',  
'\\<servername>\<sharename>\filename.bak';
```

As is the case with disk backups, if the volume hosting the fileshare fills during the backup, the backup will fail.

Performing a Full Database Backup

After you have created a logical backup device, performing a full backup to that device is relatively simple. To perform a full backup of the AdventureWorks database to the network share backup device named *networkback* that was created in an earlier example, execute the following Transact-SQL statement:

```
BACKUP DATABASE AdventureWorks  
TO networkback
```

Performing a Differential Database Backup

To back up only extents in the AdventureWorks database that have changed since the last full backup was performed on the network share backup device named *networkback* (which was created in an earlier example), execute the following Transact-SQL statement:

```
BACKUP DATABASE AdventureWorks  
TO networkback  
WITH DIFFERENTIAL
```

Performing a Transaction Log Backup

You can take transaction log backups only if the database is set to the full or bulk-logged recovery model. Recovery models are covered in Chapter 4, Lesson 2. Prior to performing a transaction log backup, first perform a full or differential backup. Once these backups have completed, you can perform a transaction log backup to the network share backup device named networkback that was created in an earlier example by executing the following Transact-SQL statement:

```
BACKUP LOG AdventureWorks  
TO networkback
```

Manual Maintenance

Sometimes it is quicker to execute a Transact-SQL command to perform a one-off index rebuild, defragmentation, or statistics update than it is to go through the process of building a maintenance plan. This section provides the Transact-SQL code used to perform some of the maintenance tasks that you can configure using the Maintenance Plan Wizard.

Check Database Integrity

As discussed in Lesson 4 of Chapter 4, the Transact-SQL code for checking a database's integrity is as follows:

```
DBCC CHECKDB
```

After it has completed running, this procedure provides a report about the state of the database and indicates what steps, if any, need to be taken next.

Reorganize Index

The Transact-SQL code used to manually reorganize an index is as follows:

```
ALTER INDEX indexname ON tablename REORGANIZE
```

Rebuild Index

The Transact-SQL code used to manually rebuild a specific index is as follows:

```
ALTER INDEX indexname ON tablename REBUILD
```

To manually rebuild all indexes on a table, use the following Transact-SQL code:

```
ALTER INDEX ALL ON tablename REBUILD;
```

PRACTICE Create a Backup Device and Take a Full Backup

The following practice creates a file backup device named *diskback* on the server Melbourne. After you create the file backup device, you will take a full backup of the AdventureWorks database.

1. Create the directory `c:\backups`.
2. Open SQL Server Management Studio on the server Melbourne, and connect to the local instance.
3. Expand the System Databases folder.
4. Right-click the master database and choose New Query.
5. In the query window, type the following Transact-SQL statement:

```
USE master;  
GO  
EXEC sp_addumpdevice 'disk', 'diskback', 'c:\backups\diskback.bak';
```

6. Click Execute. You will be informed that the command has completed successfully.
7. Right-click the AdventureWorks database and choose New Query.
8. In the query window, type the following Transact-SQL statement:

```
BACKUP DATABASE AdventureWorks  
TO diskback;
```
9. Click Execute.
10. After some time, you will be informed that BACKUP DATABASE successfully completed.
11. Examine the `c:\backups` directory. It will contain a file named `diskback.bak` that is approximately 173 MB in size.

Lesson Summary

- Maintenance plans are a simplified way of automating maintenance tasks.
- There are two ways of generating maintenance plans: using a wizard and using an IDE. Although the IDE provides more flexibility, it should be used only by experienced administrators.
- Maintenance plans can include tasks such as database integrity checks, disk space reclamation, index reorganization and rebuilding, statistics updates, history cleanups, and database backups.

- Backup devices are logical aliases for backup locations, such as local disk drives, network shares, and tape devices. SQL Server databases cannot be backed up to a tape drive attached to a different computer.

Lesson Review

You can use the following questions to test your knowledge of the information in Lesson 2, “Designing a Database Maintenance Plan.” The questions are also available on the companion CD if you prefer to review them in electronic form.

NOTE Answers

Answers to these questions and explanations of why each answer choice is correct or incorrect are located in the “Answers” section at the end of the book.

1. Which of the following maintenance tasks will compact a clustered index?
 - A. Shrink Database
 - B. Reorganize Index
 - C. Check Database Integrity
 - D. Update Statistics
2. Which of the following maintenance tasks is used to ensure that the query optimizer has up-to-date information about the distribution of data values within database tables?
 - A. Rebuild Index
 - B. Reorganize Index
 - C. Check Database Integrity
 - D. Update Statistics
3. Which of the following maintenance plan tasks allows an administrator to schedule the execution of a custom SSIS package?
 - A. Execute SQL Server Agent Job
 - B. Clean Up History
 - C. Update Statistics
 - D. Check Database Integrity

4. Which dynamic management view should you query to display all tape backup devices installed on a computer hosting a SQL Server 2005 instance?
- A. *sys.dm_io_backup_tapes*
 - B. *sys.dm_io_pending_io_requests*
 - C. *sys.dm_io_cluster_shared_drives*
 - D. *sys.dm_io_virtual_file_stats*

Lesson 3: Managing Reporting Services

Reporting Services is a set of processing components, tools, and programmatic interfaces that allow for the creation of rich reports. Reports can draw content from a variety of data sources and be published in a multitude of formats. Reports can be published through Internet Information Services (IIS), published to shared folders, or even sent by e-mail to a specific set of recipients. Reporting services allows administrators to centrally manage report security, publication, and subscriptions.

After this lesson, you will be able to:

- Create a basic report.
- Publish a basic report.
- Create a linked report.
- Use roles to differentiate access to reporting services.
- Modify report execution properties.
- Create report snapshots.
- Modify report snapshot history.
- Create and modify report subscriptions.
- Move a reporting server to another instance of SQL Server 2005.

Estimated lesson time: 30 minutes

The Report Server Database

SQL Server 2005 Reporting Services uses two databases to separate permanent data storage from temporary data storage. The default names of these databases are ReportServer and ReportServerTempDB. Report server database information can be accessed through management tools (such as Report Manager and SQL Server Management Studio) or through the Report Server Web service or Window Management Instrumentation (WMI) provider. The report server database stores the following content:

- Items managed by report server such as reports, linked reports, shared data sources, report models, folders, and resources
- Subscription and schedule definitions
- Report snapshots and history
- System properties and system-level security information

- Report execution log data
- Symmetric keys and encrypted connection information for report data sources

The temporary database stores session and execution data, cached reports, and work tables. Cached reports are also known as *temporary snapshots*. These temporary snapshots can be far larger than the report definition on which they are based. Temporary snapshots are stored within the temporary database by default, though you can configure report server to store these temporary snapshots in the file system. You can also compress temporary snapshots before they are written to disk, though this can have performance implications for the SQL Server 2005 instance. If the temporary database is dropped, you must run Setup to create a new version. You should back up the report server temporary database regularly so that if the database is lost, you do not need to re-create it from scratch.

Report Manager

Report Manager is a report access and management tool that runs within a Web browser. You access it on the computer with Reporting Services installed through the URL *localhost/Reports*. Report Manager can be used to perform the following tasks:

- View, search, and subscribe to reports.
- Create, secure, and maintain the folder hierarchy to organize items on the report server.
- Configure site properties and defaults.
- Configure the availability of My Reports.
- Configure role-based security.
- Configure report execution properties.
- Configure report parameters.
- Configure report history.
- Create report models that connect to and retrieve data from SQL Server Analysis Services data sources or from a SQL Server relational data source.
- Create shared schedules.
- Create shared data sources.
- Create data-driven subscriptions.
- Create linked reports.
- Launch Report Builder.

The default Report Manager URL is *localhost/Reports*, but you can change the virtual server directory by using the Reporting Services Configuration Manager. The tasks that a user can perform with Report Manager depend on what roles a user has been assigned. Role assignments are covered in more detail later in the lesson.

Creating a Basic Report

To demonstrate the options that you can apply to reports, such as subscriptions and report history, you will create and publish a report to the Reporting Services server. In this example, you will extract a report listing employee ID and sick leave hours from the AdventureWorks database. To create this basic report and publish it, perform the following steps:

1. Open SQL Server Management Studio.
2. Right-click the Server and select Properties.
3. Select the Security page.
4. Select the SQL Server And Windows Authentication Mode option button. Click OK. You will be instructed to restart SQL Server.
5. Right-click the server and select Restart. Click Yes to restart the server.
6. Open the Security folder and locate the Logins folder.
7. Right-click the Logins folder and select New Login.
8. Set the Login Name to **Adwrkscon**.
9. Select the SQL Server Authentication option button.
10. Clear the Enforce Password Policy, Enforce Password Expiration, and User Must Change Password At Next Login check boxes.
11. In the Password and Confirm Password fields, enter **P@ssw0rd**.
12. On the Server Roles page, set the server role to sysadmin. Normally, you would not do this, but in this example it saves time configuring permissions.
13. Click OK.
14. Open Business Intelligence Development Studio.
15. From the File menu, choose New and then Project.
16. Select Report Server Project Wizard from the Microsoft Visual Studio installed templates. Ensure that the Create Directory For Solution check box is selected. Click OK.
17. The Report Wizard starts. Click Next.

18. On the Select The Data Source page, click Edit. This opens the Connection Properties dialog box.
19. In the Server name drop-down list, select MELBOURNE.
20. In the Select Or Enter A Database Name drop-down list, select AdventureWorks as shown in Figure 6-11.

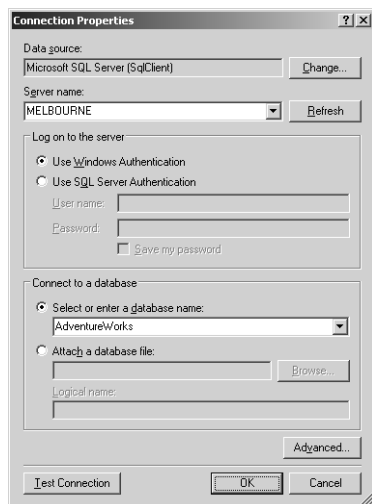


Figure 6-11 Configuring a report data source.

21. Select the Use SQL Server Authentication option.
22. Enter the user name as **Adwrkscon** and the password as **P@ssw0rd**.
23. Click Test Connection to verify that the connection works. Click OK.
24. In the Design The Query page of the Report Wizard, enter the following query string:


```
SELECT EmployeeID, SickLeaveHours FROM AdventureWorks.HumanResources.Employee ORDER BY SickLeaveHours DESC;
```
25. Click Next.
26. On the Select The Report Type page, select Tabular and then click Finish.
27. Leave the Report Name as Report1. Select the Preview Report check box, and then click Finish.
28. On the BIDS toolbar, click Save.
29. In the Solution Explorer pane of BIDS, right-click Report Project1 and choose Deploy.

30. Confirm in the Output window that the data source and report have been deployed to *localhost/ReportServer*.
31. Close BIDS.
32. Open Report Manager by navigating to *localhost/Reports/* in Internet Explorer.
33. Click Show Details. You should see items named Data Sources and Report Project1, as shown in Figure 6-12.

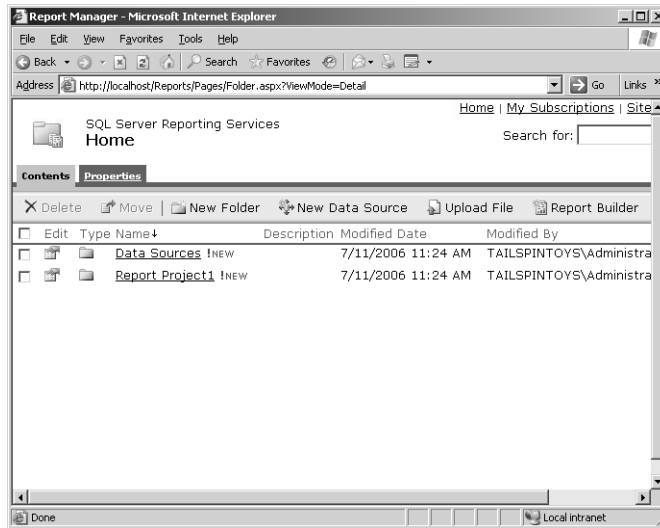


Figure 6-12 Newly published reports on the Reporting Services server.

Linked Reports

Linked reports provide access points to existing reports. Linked reports are derived from existing reports and retain the base report's definition. Linked reports inherit layout and data source properties from the base report. All other types of properties and settings—such as security, parameters, locations, subscriptions, and schedules—can be different from the base report. Linked reports are often used to create additional versions of an existing report with different settings. For example, you could use a linked report to create sales reports for each country based on an existing global sales report, or a sales report on a particular type of product from a larger sales report that includes data on all products. To create a linked report, perform the following steps:

1. Open Report Manager by navigating to *localhost/Reports*.

2. Click the report that will form the basis of the linked report. If you performed the previous task, this report will be Report Project1. Click that report. If you performed the previous task, the report that is displayed is Report1.
3. Click the Properties tab.
4. On the General Properties page for the base report, click Create Linked Report.
5. Enter a name and description for the linked report as shown in Figure 6-13.

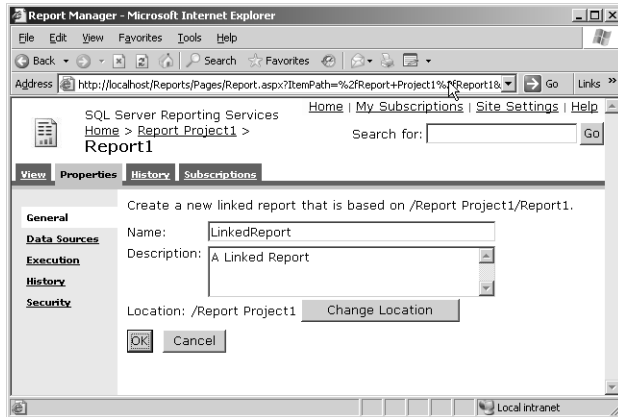


Figure 6-13 Creating a linked report based on another report.

6. Click OK. The linked report opens.

Report Execution Properties

Report execution properties control how a report is processed. You must configure execution properties for each report on an individual basis, and you cannot configure them at the server level. You configure a report's execution properties by selecting Execution on a report's Properties page in Report Manager as shown in Figure 6-14.

You can configure reports to run in the following ways:

- **On Demand** With this option, the data source is queried at the time the user runs the report. A new instance of the report is created for each user who requests the report.
- **On Demand From Cache** A report and its data are cached temporarily when a user runs a report. Other users who run the report view the cached report rather than a new report. Cached reports expire after a preconfigured amount of time.

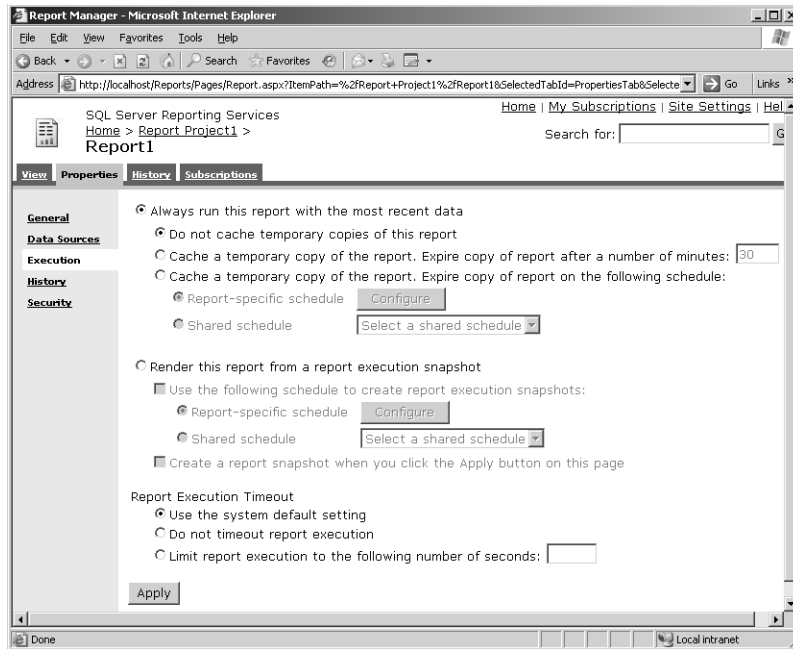


Figure 6-14 Configuring report execution properties using Report Manager.

- **From Snapshots** This option allows reports to be run against data from a specific point in time. Report snapshots are created and refreshed according to a schedule. This option can decrease the load on a server at peak times. For example, a report snapshot can be taken every morning at 2 A.M. for a report that normally has long running queries. Reports run against the snapshot rather than against the data source. Existing report snapshots are overwritten when a new snapshot is taken. Report snapshots cannot be used for reports that require user credentials or Windows Integrated security.

Report Snapshots

A report snapshot is a report made at a particular point in time. A report history is a collection of report snapshots of the same report over a period of time. You can create a report history manually or by using a schedule. Automation is limited to reports that can run unattended. The properties of a report history determine how it is created and limit the number of snapshots that are stored. You can configure properties at the server level or for individual reports. The properties you set at the server level dictate the upper limit on the number of snapshots stored in a report history. You cannot specify the limit in terms of data storage, only by number of snapshots. When the report snapshot limit is reached, older snapshots are removed to make way for new ones.

Only roles that have the Manage Report History task can create a report history. A report history can be viewed by all users whose role includes the View Reports task. Report history properties can be configured using Report Manager or SQL Server Management Studio.

You can create snapshots by opening a report in Report Manager, selecting the History tab, and clicking New Snapshot. You can accomplish the same task in SQL Server Management Studio by connecting to Reporting Services, expanding the report, right-clicking the History node, and choosing New Snapshot.

You can delete snapshots from the History page in Report Manager. In SQL Server Management Studio, you accomplish the same task by navigating to the History folder of a specific report, right-clicking a snapshot, and then choosing Delete. When you delete a report, the report history that is associated with that report is also deleted.

To configure the report history for a report server, complete the following steps:

1. Open Report Manager by navigating to *localhost/Reports* using Internet Explorer.
2. Click Site Settings in the upper right-hand corner of the screen.
3. Select the Limit The Copies Of Report History option, and enter **10** in the text box next to this option, as shown in Figure 6-15.

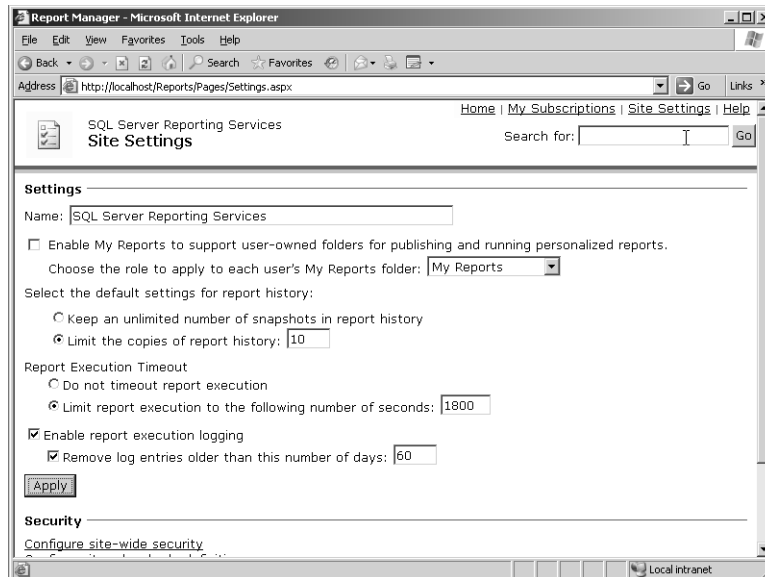


Figure 6-15 Configuring default settings for report history.

4. Click Apply.

Report Subscriptions

Subscriptions allow reports to be delivered in response to events or at specified times. Subscriptions are an alternative to manually running reports. There are two types of subscriptions: standard and data driven. *Standard subscriptions* consist of static values that cannot be varied during subscription processing. Standard subscriptions are created and managed by users. Each standard subscription has one set of presentation options, delivery options, and report parameters.

Data-driven subscriptions have their presentation, delivery, and parameter values retrieved at run time from a data source. Data-driven subscriptions are best used for large recipient lists or when you need to vary the report output for each subscriber. Data-driven subscriptions require expertise in building queries and the use of parameters. Data-driven subscriptions are almost always created by report server administrators.

Subscriptions use delivery extensions to determine the report format and the distribution method. Reporting Services supports e-mail delivery and delivery to a shared folder. Subscriptions consist of the following components:

- A report that can run unattended
- A delivery method
- A rendering extension to present the report in a particular format
- Conditions for processing the subscription
- Parameters used when running the report

Subscription information is stored with individual reports in the report server database. You cannot manage subscriptions separately from reports. To configure a standard subscription schedule, perform the following steps:

1. In SQL Server Configuration Manager, ensure that the SQL Server Agent service is started.
2. Create a share named **reports** on the local computer.
3. Open Report Manager by using Internet Explorer to navigate to *localhost/Reports*.
4. Click the report for which you want to configure a schedule.
5. Click the View tab.
6. Click New Subscription.

7. In the Report Delivery Options drop-down list, select Report Server File Share.
8. Set the Path field to *localhost\reports*.
9. Specify the appropriate values for User Name and Password. In the real world, this would be something secure, but in this lab you can use the credentials you used to log in to the server.
10. Select the Increment File Names As Newer Versions Are Added overwrite option.
11. Click Select Schedule.
12. Configure the schedule to run at 8:00 A.M. on Monday, Wednesday, and Friday.
13. Click OK.
14. Enter a password in the Password dialog box because this box will have been cleared when you configured a schedule. Click OK.

Reporting Services Configuration Manager

You can use the Reporting Services Configuration Manager, shown in Figure 6-16, to configure Reporting Services on SQL Server 2005. You start the Reporting Services Configuration Manager from the Configuration Tools menu of the Microsoft SQL Server 2005 programs group.

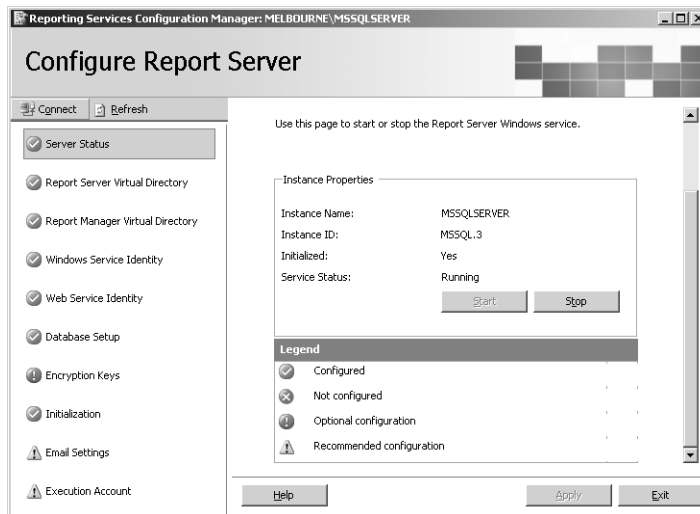


Figure 6-16 Reporting Services Configuration Manager.

You use the Reporting Services Configuration Manager to perform the following tasks:

- **Create and configure virtual directories.** Report Server and Report Manager are ASP.NET applications that you access through a Web browser. You use Reporting Services Configuration Manager to configure the location of these virtual directories.
- **Configure service accounts.** You can use this tool to manage the service accounts that are used to run the Report Server Web service and the Report Server Windows service. To configure these accounts, select either Windows Service Identity or Web Service Identity, respectively. You should use the Reporting Services Configuration Manager, rather than the Services console, to alter Report Server accounts because it also automatically updates encryption keys and profile information, which is something the Services console cannot do.
- **Create and configure connections to the report server database.** The tool can be used to create or select a Report Server database as long as it uses the SQL Server 2005 Reporting Services schema.
- **Manage encryption keys and initialization.** The tool can be used to back up, restore, and re-create the encryption keys used to encrypt and decrypt data such as credentials and database connection information.
- **Configure e-mail delivery.** The tool can be used to specify which SMTP server should be used for e-mail delivery.
- **Configure the scaling out of the Report Server deployment model.** Configure multiple Report Server instances to use a single, shared Report Server database.

If a report server is to generate reports based on data stored on other servers within the same domain, you will need to configure the SQL Server Agent and Report Server Windows services to run under a domain-based account rather than a local account.

Quick Check

1. What are the three ways that you can configure reports to run?
2. What are the two types of subscriptions?

Quick Check Answer

1. Reports can be configured to run on demand, on demand from cache, and from snapshots.
2. The two types of subscriptions are standard and data driven.

Configuring Role-Based Security

SQL Server 2005 Reporting Services includes a number of predefined roles that you can use to control which tasks a user can and cannot perform. As a DBA, you can modify these roles or replace them with custom roles. The five predefined Reporting Services roles and two system roles are as follows:

- Browser
- Content Manager
- Report Builder
- Publisher
- My Reports
- System Administrator
- System User

BEST PRACTICES Creating custom roles

If you need to grant users abilities that are different from those granted by the predefined roles, it is better to create custom roles than to modify the existing ones. If you leave your position in the company, the person taking over your job will expect the predefined roles to work in the default way. Only if your replacement is diligent will she check to see whether these roles have been customized.

Browser Role

Users assigned to the Browser role can view reports but are unable to make any changes to reports. The following tasks are included in the Browser role definition:

- **View Reports** This task is used for running a report and viewing report properties.
- **View Resources** This task is used to view resources and resource properties.
- **View Folders** This task enables the user to view folder contents and navigate the folder structure.
- **View Models** This task allows the user to view models, use models as data sources, and run queries against models.
- **Manage Individual Subscriptions** This task allows the user to view, create, delete, and modify user-owned subscriptions, as well as create schedules to support those subscriptions.

Content Manager Role

The Content Manager role is useful for users who manage reports or Web content but do not author reports, manage IIS, or manage SQL Server itself. Content managers deploy reports, manage data source connections, and manage report models. Administrators should assign this role to trusted users only because an uploaded report or HTML file can contain malicious scripts. If a published report contains a malicious script, any user who executes that report will execute the malicious script when the report is opened using her credentials. All tasks from the Browser role, as well as the following tasks, are included in the Content Manager role definition:

- **Consume Reports** This task enables a user to read report definitions.
- **Create Linked Reports** This task enables a user to create link reports based on non-linked reports.
- **Manage All Subscriptions** This task enables a user to view, modify, and delete report subscriptions for any users.
- **Manage Data Sources** This task enables a user to create, delete, and modify data source items.
- **Manage Folders** This task enables a user to create, delete, and modify folders.
- **Manage Models** This task enables a user to create, delete, and manage models.
- **Manage Report History** This task enables a user to create, view, modify, and delete report history and settings that determine snapshot history limits and caching.
- **Manage Reports** This task enables a user to add and delete reports. It also allows a user to modify report parameters, properties, definitions, and security policies, as well as to view and modify data sources that provide content to a report.
- **Manage Resources** This task enables a user to create, modify, and delete resources. It also allows a user to view and modify resource properties.
- **Set Security Policies For Items** This task enables a user to define security policies for items.

Report Builder Role

The Report Builder role is almost identical to the Browser role except that it has the Consume Reports task, which allows a user assigned this role to load a report definition

from the report server into a local Report Builder instance. Report Builder is a client application that is used to process a report without using a report server.

Publisher Role

The Publisher role is assigned to users who need to add content to a report server. It is intended for users who author reports in Report Designer and publish those reports to a report server. As with the Content Manager role, this role allows users to upload any type of file to a report server, and care should be taken to ensure that malicious scripts are not published deliberately or inadvertently. The Publisher role allows a subset of the tasks of the Content Manager role. The tasks included in this role are Create Linked Reports, Manage Data Sources, Manage Folders, Manage Reports, Manage Models, and Manage Resources. Each of these tasks is described in the “Content Manager Role” section earlier in this lesson.

My Reports Role

This role allows users to manage report stores within their My Reports folder. Users who are assigned this role can perform the following tasks within the My Reports folder that they own:

- Create Linked Reports
- Manage Folders
- Manage Data Sources
- Manage Individual Subscriptions
- Manage Reports
- Manage Resources
- View Reports
- View Data Sources
- View Resources
- View Folders

These tasks are described in the sections on the Browser and Content Manager roles, but they can be performed only on items in the assigned My Reports folder.

System Administrator Role

The System Administrator role is assigned to users who have responsibility for a report server rather than the content it hosts. A user assigned the System Administrator role can perform the following tasks:

- **Execute Report Definitions** This task enables a user to start the execution of a report definition without publishing the report definition to a report server.
- **Manage Jobs** This task enables a user to view and terminate running jobs.
- **Manage Report Server Properties** This task enables a user to view and modify report server properties, as well as view and modify items that the report server manages. With this task, a user can rename Report Manager, enable My Reports, and set report history defaults.
- **Manage Roles** This task enables a user to create, view, modify, and delete role definitions using the Site Settings page.
- **Manage Shared Schedules** This task enables a user to create, view, modify, and delete shared schedules.
- **Manage Report Server Security** This task enables a user to view and modify system-wide role assignments.

System User Role

The System User role allows users to view basic information about the report server. It provides access to fewer tasks than any other role, including the Browser role. Rather than viewing information about reports, the System User role allows the viewing of basic information about the report server itself. This role also includes support for loading reports using Report Builder, though it does not allow access to as many tasks as the Report Builder role does. Users assigned to the System User role can perform the following tasks:

- **Execute Report Definitions** With this task, users can start execution for a report definition without publication to a report server.
- **View Report Server Properties** With this task, users can view properties of the report server, such as application name, report history defaults, and whether My Reports is enabled.
- **View Shared Schedules** With this task, users can view shared schedules that are used to run reports.

Moving a Report Server

If Report Server degrades the performance of an instance so much that users begin to complain, a DBA can move the Report Server to another instance on a separate computer. Moving a database does not influence scheduled operations currently defined for Report Server items. All schedules are re-created when the Report Server service is started on the destination instance. This is because subscriptions, cached reports, and snapshots are preserved in the moved database.

There are three possible methods for moving report server databases: backup and restore, attach and detach, and copy. The approach that a DBA implements varies depending on availability requirements. Attaching and detaching provides the easiest method, but the report server will be offline during the process. Backup and restore minimizes the impact on availability, but it requires complex Transact-SQL commands during the operation. Using the Copy Database Wizard is fast, but it does not preserve database permission settings. After you have moved a database, you must reconfigure the Report Server connection to the Report Server database. To move the Report Server database by using the detach and attach method, follow these steps:

1. Stop the Report Server Windows service and the Web application pool that hosts the Report Server Web service.

NOTE Choosing a shut-down method

The delicate way to perform this operation is to shut down the Web application pool that hosts the Report Server Web service. The less delicate way is to shut down IIS entirely.

2. Open a connection in SQL Server Management Studio to the SQL Server instance hosting the report server database.
3. Right-click the ReportServer database, choose Tasks, and choose Detach.
4. Right-click the ReportServerTempDB database, choose Tasks, and choose Detach.
5. Transfer the .mdf and .ldf files to the Data folder of the target SQL Server instance. Four files in total should be transferred.
6. Use SQL Server Management Studio to connect to the target instance.
7. Right-click the Databases folder and choose Attach.
8. Click Add to select the report server database files that you transferred in step 5. Repeat this step for the ReportServerTempDB files.
9. Verify that the RSExecRole is a database role and has the Select, Create, Update, Delete, and References permissions on all tables in the ReportServer and

ReportServerTempDB databases as shown in Figure 6-17. Ensure that RSExecRole has execute permissions on the stored procedures.

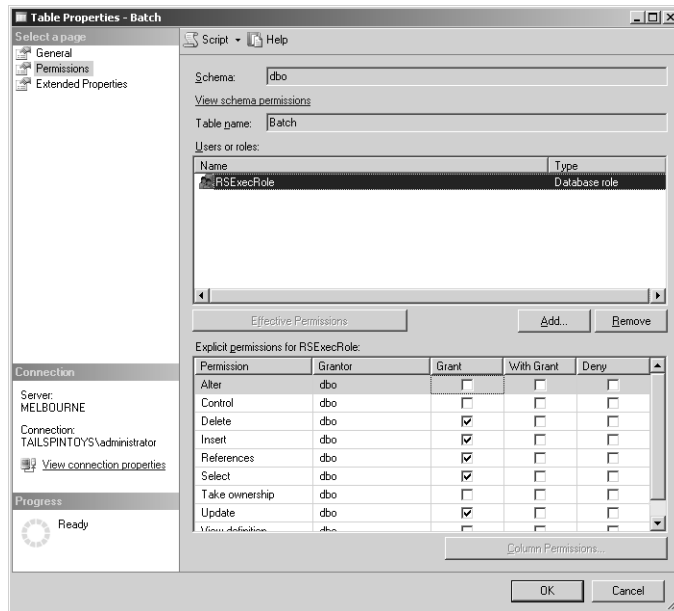


Figure 6-17 Verifying that RSExecRole has the correct table permissions.

- From the Configuration Tools menu of the Microsoft SQL Server 2005 program group, start the Reporting Services Configuration tool and open a connection to the destination instance as shown in Figure 6-18.

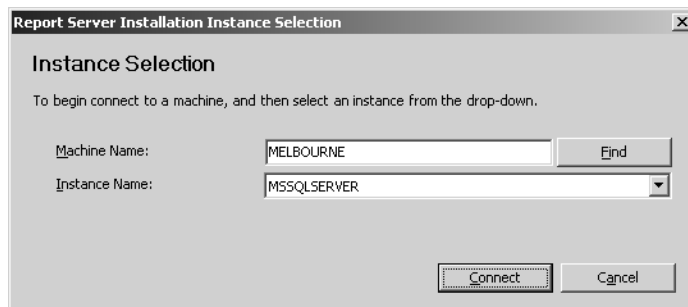


Figure 6-18 Opening a connection to the destination instance.

- On the Database Setup page, select the destination SQL Server instance, and click Connect.
- Select the report server database that you just transferred, and click Apply.
- Restart the Report Server Windows service and the Web application pool on the original server.

PRACTICE Generate a Report

In the following practice, you generate a report that can be used by the sales team at AdventureWorks. The report displays information about the highest earning customers that AdventureWorks has.

1. Open Business Intelligence Development Studio.
2. From the File menu, choose New and then Project.
3. Select the Report Server Project Wizard from the Visual Studio installed templates.
4. Enter the project name as **RichCustomers**, and ensure that the Create Directory For Solution check box is selected. Click OK.
5. When the wizard starts, click Next.
6. On the Select The Data Source page, verify that New Data Source is selected. Enter the data source name as **AdWorks**.
7. Click Edit. In the Server Name drop-down list, select Melbourne.
8. In the Select Or Enter A Database Name drop-down list, select AdventureWorks.
9. Click Test Connection.
10. Click OK twice.
11. Click Credentials.
12. Verify that Use Windows Authentication is selected. Click OK.
13. Click Next.
14. In the Query String text box, enter the following string:

```
SELECT FirstName, LastName, Phone, YearlyIncome FROM
AdventureWorks.dbo.HighIncomeCustomers ORDER BY YearlyIncome DESC;
```
15. Click Next.
16. Select Tabular and click Next.
17. Click Next again.
18. On the Table Style page, choose Ocean.
19. Click Next.
20. Leave the default values on the Deployment Location page, and click Next.
21. Enter the report name as **Rich Customers**, and click Finish.
22. In Solution Explorer, right-click RichCustomers and select Deploy.

23. Verify in the Output window that the project has successfully deployed.
24. Open Report Manager in Internet Explorer, and verify that the new report is present.

Lesson Summary

- Reporting services uses two databases to separate permanent data storage from temporary data storage.
- Report server database information can be accessed through Report Manager or SQL Server Management Studio.
- The report server database stores reports, linked reports, shared data sources, report models, folders, subscriptions, schedule definitions, report snapshots, and history.
- The functionality of the Report Manager depends on which roles have been assigned to a user.
- Linked reports are derived from existing reports and retain the base report's definition.
- A report snapshot is a report created at a particular point in time. A report history is a collection of report snapshots.
- Report subscriptions allow reports to be delivered in response to events or at specified times.
- SQL Server 2005 reporting services ships with five predefined reporting services roles and two system roles. These roles are the Browser role, Content Manager role, Report Builder role, Publisher role, My Reports role, System Administrator role, and System User role.

Lesson Review

You can use the following questions to test your knowledge of the information in Lesson 3, "Managing Reporting Services." The questions are also available on the companion CD if you prefer to review them in electronic form.

NOTE Answers

Answers to these questions and explanations of why each answer choice is correct or incorrect are located in the "Answers" section at the end of the book.

1. Which of the built-in reporting services roles are allowed to perform the Create Linked Reports task outside of the My Reports folder? (Choose all that apply.)
 - A. Content Manager
 - B. Publisher
 - C. Browser
 - D. Report Builder
2. You need to assign several Reporting Services users the ability to create resources outside the My Reports folder. Which of the following roles could you assign to these users?
 - A. Publisher
 - B. System User
 - C. My Reports
 - D. Browser
3. You are logged on to a SQL Server computer that has Reporting Services installed. You want to use Internet Explorer to connect to the local instance of Report Manager. Which of the following URLs should you use?
 - A. *localhost/ReportManager*
 - B. *localhost/ReportServer*
 - C. *localhost/Reports*
 - D. *localhost/*

Lesson 4: Designing a Strategy to Manage Data Across Linked Servers

This lesson concentrates on configuring and managing SQL Server 2005 linked server configurations and queries. Linked server configurations allow SQL Server 2005 to execute commands against OLE DB data sources on other servers. OLE DB is the successor to ODBC and allows access to data sources as diverse as Oracle, DB2, and Microsoft Office Excel.

After this lesson, you will be able to:

- Set up linked servers.
- Manage linked servers.
- Specify login security.
- Define link locations and security.

Estimated lesson time: 30 minutes

Linked Server Basics

Linked servers usually handle distributed queries. A *distributed query* is a query that accesses data from multiple heterogeneous data sources. SQL Server 2005 supports distributed queries by using OLE DB, a replacement technology for ODBC. Because they use OLE DB, you can use distributed queries to access the following information from an instance of SQL Server 2005:

- Data stored in multiple instances of SQL Server
- Heterogeneous data stored in many different types of data sources that allow access via an OLE DB provider

NOTE Heterogeneous data sources

The term *heterogeneous data* is another way of saying “data stored in multiple formats”—for example, it can be used to describe data stored on a SQL Server database, on an open source database, in an access database, or within an Excel spreadsheet.

How Linked Servers Work

Figure 6-19 provides a basic outline of the linked server process. Linked servers activate when a client application executes a distributed query against the linked

server. The SQL Server 2005 database engine examines the command and forwards the requests to OLE DB. OLE DB then uses the OLE DB providers to communicate with data sources—usually other database servers such as Oracle, Microsoft Access, DB2, and other SQL Server 2005 computers—to resolve the distributed query.

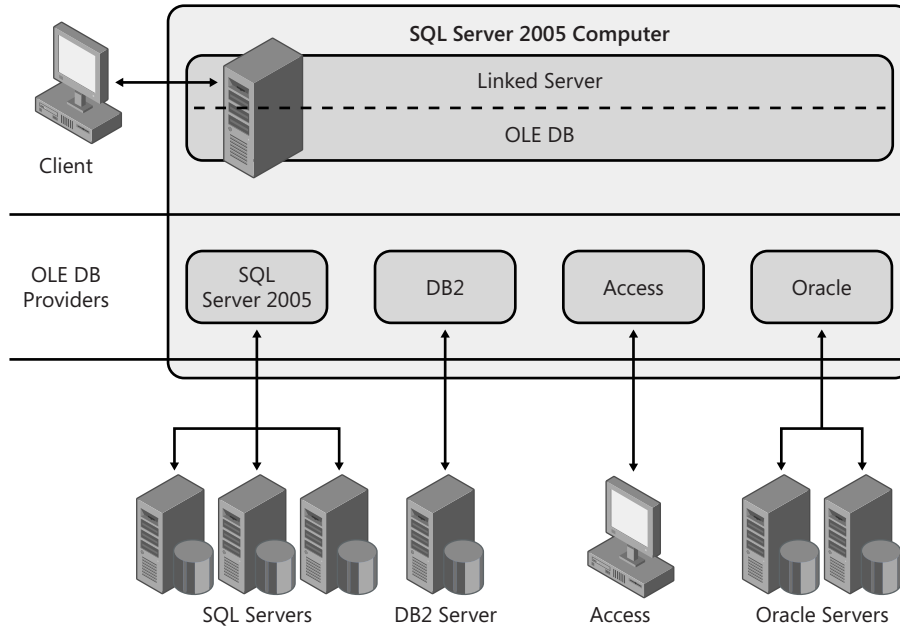


Figure 6-19 Overview of a linked server configuration.

Setting Up Linked Servers

A linked server definition requires both an OLE DB provider and an OLE DB data source. OLE DB providers manage specific data sources. OLE DB data sources are used to specify databases that can be accessed using OLE DB. In general, the data sources queried through linked server definitions are other databases. OLE DB providers also exist for other file formats, such as text files and Excel spreadsheet data.

MORE INFO Tested OLE DB providers

For a list of all OLE DB providers tested with SQL Server 2005, consult the following MSDN article: msdn.microsoft.com/library/default.asp?url=/library/en-us/acdata/ac_8_qd_12_15ma.asp.

The system stored procedure you use to create linked servers is *sp_addlinkedserver*. After you have created the linked server, you can run distributed queries against it. If you define the linked server as an instance of SQL Server 2005, you can also execute remote stored procedures.

The *sp_addlinkedserver* stored procedure has the following syntax:

```
sp_addlinkedserver [ @server= ] 'server' [ , [ @srvproduct= ] 'product_name' ]
    [ , [ @provider= ] 'provider_name' ]
    [ , [ @datasrc= ] 'data_source' ]
    [ , [ @location= ] 'location' ]
    [ , [ @provstr= ] 'provider_string' ]
    [ , [ @catalog= ] 'catalog' ]
```

A description of the arguments for *sp_addlinkedserver* are provided in Table 6-1.

Table 6-1 Argument Descriptions for *sp_addlinkedserver*

Argument	Description
@server=	Name of the linked server to create. Server is sysname.
@srvproduct=	OLE DB data source product name to be added as a linked server. If this field is set to SQL Server, provider_name, data_source, location, provider_string, and catalog need not be specified.
@provider=	Unique programmatic identifier (PROGID) of the OLE DB provider. Provider_name must be unique for the OLE DB provider installed on instance. If provider_name is omitted, SQLNCLI (SQL native client) is used. The OLE DB provider is registered with PROGID in the registry.
@datasrc=	Data source name as interpreted by the OLE DB provider.
@location=	Database location as interpreted by OLE DB provider.
@provstr=	The OLE DB provider-specific connection string that identifies a unique data source.
@catalog=	Catalog to be used when a connection is made to OLE DB provider. Catalog is sysname, with a default of NULL.

Following are three separate examples of linked server creation. The first example shows the statement used to create a linked server titled LinkServ1 on an instance of SQL Server. LinkServ1 uses the SQL Native Client OLE DB (SQLNCLI).

```
EXEC sp_addlinkedserver
    @server='LinkServ1',
    @srvproduct='',
    @provider='SQLNCLI',
    @datasrc='TargetServer\TargetInstance'
```

The second statement creates a linked server named Melbourne HR. This statement is more complex because it uses the Microsoft OLE DB Provider for Oracle. The statement assumes that the SQL*Net alias for the Oracle database is LocalServer.

```
EXEC sp_addlinkedserver
    @server = 'Melbourne HR',
    @srvproduct = 'Oracle',
    @provider = 'MSDAORA',
    @datasrc= 'LocalServer'
```

The final statement creates a linked server named DB2 for a DB2 database named DB2-Database. This linked server uses the Microsoft OLE DB provider for DB2, DB2OLEDB. You need to provide far more information to configure this linked server than to configure a linked server to another SQL Server 2005 instance.

```
EXEC sp_addlinkedserver
    @server='DB2',
    @srvproduct='Microsoft OLE DB Provider for DB2',
    @catalog='DB2',
    @provider='DB2OLEDB',
    @provstr='Initial Catalog=DB2-DatabaseName;
    Data Source=DB2;
    HostCCSID=1252;
    Network Address=W.X.Y.Z;
    Network Port=50000;
    Package Collection=admin;
    Default Schema=admin;'
```

MORE INFO *sp_addlinkedserver*

For more information about the *sp_addlinkedserver* stored procedure, consult the following MSDN article: [msdn2.microsoft.com/en-us/library/ms190479\(d=ide\).aspx](http://msdn2.microsoft.com/en-us/library/ms190479(d=ide).aspx).

Configuring OLE DB Provider Options

You can use SQL Server Management Studio to configure OLE DB provider options. To configure OLE DB provider options, perform the following steps:

1. Open SQL Server Management Studio, and connect to the appropriate server instance.
2. Expand the Server Objects folder.
3. Expand the Linked Servers folder.
4. Expand the Providers folder.
5. Right-click the provider you want to configure and choose Properties. This opens the Provider Options dialog box as shown in Figure 6-20.

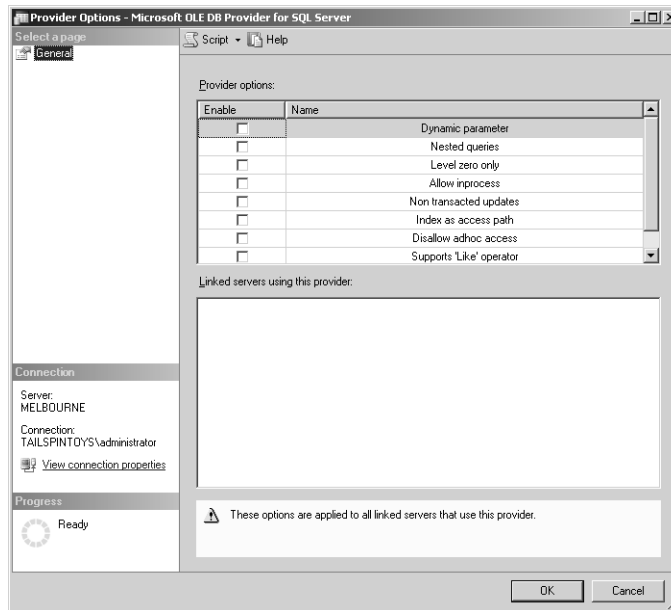


Figure 6-20 Microsoft OLE DB Provider for SQL Server options.

A list of linked servers that are currently configured to use a provider are displayed in the bottom right-hand corner of the Provider Options dialog box. Options configured for each provider apply to all linked servers that use that provider. A description of each of the provider options is given in Table 6-2.

Table 6-2 OLE DB Provider Options

Provider Option	Description
DynamicParameters	Allows the execution of parameterized queries against the provider. Can produce better performance for certain queries.
NestedQueries	Allows for nested SELECT statements in the FROM clause.
LevelZeroOnly	Only level 0 OLE DB interfaces are invoked against the provider.
AllowInProcess	Allows for the provider to be instantiated as an in-process server. The SQL Native Client OLE DB provider cannot be instantiated out of process.
NonTransactedUpdates	Updates against the provider are nonrecoverable because the provider does not support transactions.
IndexAsAccessPath	SQL Server attempts to use provider indexes to retrieve data.
DisallowAdhocAccess	Controls ability of nonadministrators to run ad hoc queries through the OPENROWSET and OPENDATASOURCE functions.
SqlServerLike	Provider supports the LIKE operator as implemented by SQL Server 2005.

CAUTION Altering OLE DB provider options

Only experienced systems administrators should alter OLE DB provider options.

Configuring Linked Servers for Delegation

You can configure SQL Server and the Windows operating system so that a client connected to one instance of SQL Server can connect to another instance of SQL Server by forwarding the credentials of that authenticated Windows user. The term for this process is *delegation*. During delegation, the instance to which the Windows authenticated user has connected impersonates that user when contacting other

instances. This process is shown in Figure 6-21. *Self-mapping* is when the currently used security credentials are emulated for resolving queries against linked servers. When a linked server is added to an instance using `sp_addlinkedserver`, self-mapping is added by default for all local logins. If the linked server supports Windows Authentication, self-mapping for all Windows logins is supported.

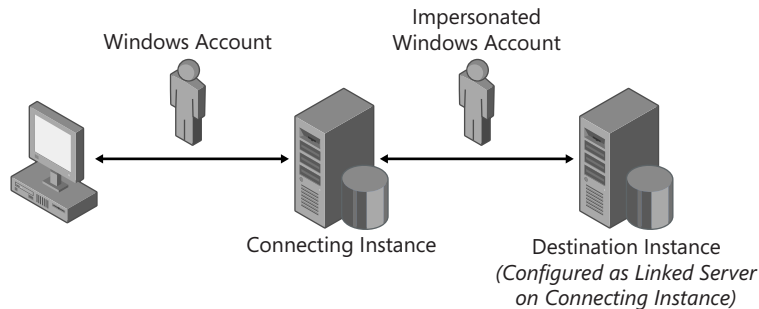


Figure 6-21 During delegation, the server a user connects to impersonates that user when connecting to other servers to resolve distributed queries.

Double hop is the term used to describe when a user logs in to a client computer that connects to the first instance and runs a distributed query against a database on a linked server, which is a second separate instance.

To participate in a double-hop configuration, the following conditions must be established:

- Both client connecting instances and destination instances must be connected via a TCP/IP network.
- The user's Windows authenticated account must have access permissions on connecting and destination instances.
- Both connecting and destination instances must have Service Principal Names (SPNs) registered by the domain administrator.
- The account that the connecting instance runs under must be trusted for delegation.
- The destination instance must be added as a linked server on the connecting instance (using the `sp_addlinkedserver` stored procedure).
- The linked server logins must be configured for self-mapping (accomplished using the `sp_addlinkedsrvlogin` stored procedure).

Linked Server Security

Although accounts that use Windows Authentication are automatically self-mapped when you create a linked server, in some circumstances you need to create a login mapping between linked servers using stored procedures. You add login mappings by using *sp_addlinkedrvlogin* and drop them by using *sp_droplinkedrvlogin*. Login mapping sets up a remote login and password for the specified linked server and connecting instance login. When the connecting instance connects to a linked server to resolve a distributed query, the instance checks whether any login mappings exist for the current login. If a login mapping is found, the connecting instance transmits the remote login and password to the linked server. As mentioned earlier, the default configuration is self-mapping, emulating the credentials of the current user to connect to the linked server. The least administratively complex way of handling linked server security is to ensure that Windows authentication is used for all instances.

Quick Check

1. What are the user account requirements for a double hop?
2. Which stored procedures are used to add and drop login mappings?

Quick Check Answer

1. The user's Windows authenticated account must have access permissions on connecting and destination instances. The account that the connecting instance runs under must be trusted for delegation. The linked server logins must be configured for self-mapping (accomplished using *sp_addlinkedrvlogin*).
2. Login mappings are added using *sp_addlinkedrvlogin* and dropped using *sp_droplinkedrvlogin*.

Self-mapping does not work if security account delegation is unavailable or if the connecting or destination instances are not configured for Windows Authentication mode. Administrators should review the Account Is Sensitive And Cannot Be Delegated and Account Is Trusted For Delegation settings when self-mapping does not work. (You can review these settings by examining the user account's properties within Active Directory Users And Computers.) If self-mapping is not possible, SQL Server authentication and server login mappings are necessary.

Configuring Linked Server Options

You use the *sp_serveroption* system stored procedure to configure server-level options rather than provider-level options. This configuration allows you to configure two different linked servers that use the same OLE DB provider to behave differently. Table 6-3 describes linked server options.

Table 6-3 Linked Server Options

Linked Server Option	Description
Use Remote Collation	When this option is set to true, SQL Server uses collation information of character columns from the linked server. This is useful when multiple databases on the remote server use different collations.
Collation Name	Specifies collation that is used for character data from the linked server if the Use Remote Collation option is set to true and the data source is not a SQL Server data source. Do not set if the linked server has multiple collations within a single data source. This option is ignored if the linked server is SQL Server 2005.
Connection Timeout	Time-out value in seconds for the connection to the linked server. If this option is not set, the global configuration option remote login time-out is the default.
Lazy Schema Validation	If this option is set to false, SQL Server checks for schema changes that have occurred since compilation in remote tables. The check occurs prior to query execution. If a change has occurred, SQL Server recompiles the query with new schema. If this option is set to true, schema checking does not occur until execution.

MORE INFO Configuring linked server options

For more information about configuring linked server options, consult the following MSDN article: [msdn2.microsoft.com/en-us/library/ms178532\(d=ide\).aspx](https://msdn2.microsoft.com/en-us/library/ms178532(d=ide).aspx).

Lesson Summary

- Linked servers are most often used to handle distributed queries.
- Distributed queries are queries that access data from multiple heterogeneous data sources.
- SQL Server supports distributed queries using OLE DB.
- Linked-server definitions require both an OLE DB provider and an OLE DB data source.
- Delegation is the process by which SQL Server forwards the credentials of an authenticated Windows user.
- Self-mapping is when the currently used credentials are emulated for resolving queries against linked servers.
- Login mappings are added using *sp_addlinkedsrvlogin* and dropped using *sp_droplinkedsrvlogin*.

Lesson Review

You can use the following questions to test your knowledge of the information in Lesson 4, “Designing a Strategy to Manage Data Across Linked Servers.” The questions are also available on the companion CD if you prefer to review them in electronic form.

NOTE Answers

Answers to these questions and explanations of why each answer choice is correct or incorrect are located in the “Answers” section at the end of the book.

1. Which of the following conditions are necessary to support self-mapping? (Choose all that apply.)
 - A. The destination instance is configured to support Windows authentication.
 - B. The connecting instance is configured to support SQL Server authentication.
 - C. The connecting instance is configured to support Windows authentication.
 - D. The destination instance is configured to support SQL Server authentication.

2. Three users of the system for which you are the administrator are able to execute distributed queries through a linked-server configuration. Another user is unable to. All four users authenticate with the connecting server using Windows authentication. Login mappings use default settings, and no login mappings have been configured for any users. Which of the following is the most likely reason that only the fourth user is unable to execute distributed queries against the linked server?
- A. The destination instance does not support Windows authentication.
 - B. The destination instance does not support SQL Server authentication.
 - C. The Account Is Trusted For Delegation account property is set in Active Directory.
 - D. The Account Is Sensitive And Cannot Be Delegated account property is set in Active Directory.

Chapter Review

To further practice and reinforce the skills you learned in this chapter, you can complete the following tasks:

- Review the chapter summary.
- Review the list of key terms introduced in this chapter.
- Complete the case scenarios. These scenarios set up real-world situations involving the topics of this chapter and ask you to create a solution.
- Complete the suggested practices.
- Take a practice test.

Chapter Summary

- DDL triggers can be configured to protect the database from user damage.
- Job dependency diagrams provide an easy-to-use way of planning all the necessary steps involved in database maintenance tasks.
- All hotfixes, security updates, and service packs should be rigorously tested on configurations similar to that of production systems prior to being applied to production systems.
- Prior to installing hotfixes, security updates, and service packs, perform a full operating system backup, a full Automated System Recovery backup, and a full backup of all databases, including system databases.
- Maintenance plans are a simplified way of automating maintenance tasks.
- Maintenance plans can include tasks such as database integrity checks, disk space reclamation, index reorganization and rebuilding, statistics updates, history cleanups, and database backups.
- The functionality of the report manager depends on which roles have been assigned to a user.
- Linked reports are derived from existing reports and retain the base report's definition.
- A report snapshot is a report created at a particular point in time. A report history is a collection of report snapshots.
- Report subscriptions allow reports to be delivered in response to events or at specified times.

- Linked servers are most often used to handle distributed queries.
- Distributed queries are queries that access data from multiple heterogeneous data sources.
- Delegation is the process by which SQL Server forwards the credentials of an authenticated Windows user.

Key Terms

Do you know what these key terms mean? You can check your answers by looking up the terms in the glossary at the end of the book.

- collation
- data definition language (DDL)
- DDL trigger
- delegation
- differential backup
- distributed query
- double hop
- Open Database Connectivity (ODBC)
- OLE DB
- report definition
- report snapshot
- self mapping

Case Scenarios

In the following case scenarios, you will apply what you've learned in this chapter. You can find answers to these questions in the "Answers" section at the end of this book.

Case Scenario 1: Managing Updates

You are in the middle of planning a update management infrastructure for your company. A consultant has been talking to one of your managers and giving her some advice about your project. With this new information in mind, your manager has some questions that she would like you to answer.

1. Why should automatic updates be disabled on production computers that run SQL Server 2005?
2. Which product should be placed on the company's screened subnet to minimize the amount of data downloaded from Microsoft update servers?
3. Which tool can be used to check whether all currently available updates, hot-fixes, and service packs have been applied to the operating system that hosts SQL Server 2005?

Case Scenario 2: Configuring Report Server Roles

You are responsible for assigning report server roles to employees at Contoso Corporation. You must assign roles to three employees. Each employee's job responsibilities are as follows:

- Don Hall needs to be able to publish reports to the report server. Don should not be able to manage subscriptions.
- Jay Hamlin needs to be able to load report definitions from the report server to the local report builder instance. Jay also needs to be able to view reports, folders, and models.
- Darren Parker needs to be able to publish reports and manage subscriptions.

With these requirements in mind, answer the following questions:

1. Which of the built-in report server roles should you assign to Don?
2. Which of the built-in report server roles should you assign to Jay?
3. Which of the built-in report server roles should you assign to Darren?

Suggested Practices

To help you successfully master the exam objectives presented in this chapter, complete the following practice tasks:

- **Practice 1: Create a job dependency diagram.** Create a job dependency diagram for an inventory database in which all data from the inventory table should be archived each month to another table named InventoryArchive and then deleted from the inventory table. The index on the InventoryArchive table should then be rebuilt.

- **Practice 2: Create a database maintenance plan for the AdventureWorks database.**
Use the Database Maintenance Wizard to create a maintenance plan for the AdventureWorks database. Once every week, the database should be subjected to an integrity check, all job history over four weeks of age should be purged, and the database should be fully backed up.
- **Practice 3: Create and publish a report.** Create and publish a report that lists employee ID, salary, and payfrequency as stored in the AdventureWorks database.
- **Practice 4: Create a linked server definition.** Create a linked server definition for an Oracle server called **Auckland**. The SQL*Net alias for this Oracle database is KiwiServer.

Take a Practice Test

The practice tests on this book's companion CD offer many options. For example, you can test yourself on just one exam objective, or you can test yourself on all the 70-444 certification exam content. You can set up the test so that it closely simulates the experience of taking a certification exam, or you can set it up in study mode so that you can look at the correct answers and explanations after you answer each question.

MORE INFO Practice tests

For details about all the practice test options available, see "How to Use the Practice Tests" in this book's Introduction.

Chapter 7

SQL Server Integration Services

SQL Server Integration Services (SSIS) is a new subsystem of SQL Server 2005 that replaces SQL Server 2000 Data Transformation Services. SSIS packages are commonly used to merge data from heterogeneous data sources, populate data warehouses, clean and standardize data, build business intelligence into a data transformation process, and automate administrative functions. In this chapter, you learn how to use the Business Intelligence Development Studio (BIDS) to design and construct SSIS packages. You also learn how to use the interface to secure sensitive data stored in packages, troubleshoot package execution using checkpoints, and deploy completed SSIS packages to other servers.

Exam objectives in this chapter:

- Design and manage SQL Server Integration Services (SSIS) packages.
 - Construct complex SSIS packages.
 - Design security for accessing packages.
 - Restart failed packages.
 - Troubleshoot or debug packages.
 - Deploy and move packages.
 - Schedule package execution.
 - Move packages to different servers.

Lessons in this chapter:

- Lesson 1: Constructing SSIS Packages. 369
- Lesson 2: Securing SSIS Packages 386
- Lesson 3: Troubleshooting SSIS Packages. 395
- Lesson 4: Deploying SSIS Packages 405

Before You Begin

To complete the lessons in this chapter, you must have completed the following tasks:

- Configured a Microsoft Windows Server 2003 R2 computer with SQL Server 2005 Enterprise Edition SP1 as detailed in the Appendix.
- Obtained an updated copy of the AdventureWorks sample database and installed it as detailed in the Appendix.

Real World

Orin Thomas

There are always a few people needing their own unique set of information either extracted from or put into the database. In the past, I'd usually spend some hours cobbling together a script that automates this sort of work. Each script I wrote was different because every person coming to me needed something unique. Because I wrote scripts only on an irregular basis, I'd have to spend a couple of hours with my nose in a book to figure out the convoluted syntax necessary to get the job done. For me, the release of SSIS with SQL Server 2005 simplified matters greatly. The Business Intelligence IDE allows me to put together packages to extract or import the appropriate data from databases in a fraction of the time it would take me to write a script to achieve the same result. Once you learn how to use SSIS packages, you'll appreciate the time and effort that they can save.

Lesson 1: Constructing SSIS Packages

In this lesson, you get an introduction to the BIDS interface, learn more about SSIS packages, and learn how you can use the BIDS interface to construct SSIS packages. Be aware that SSIS package construction and design is a complex topic worthy of a book itself. Although this chapter provides an overview of the important aspects of SSIS packages, the best way for you to become familiar with the process is by doing it yourself. After you've read through each lesson, you should practice with the exercises in this chapter and then move on to the SSIS tutorials available in SQL Server 2005 Books Online. When you are confident enough, you should use BIDS to construct a few of your own packages. Although the basics are presented in this lesson, it is through constructing your own packages that you'll best come to terms with this exam objective.

After this lesson, you will be able to:

- Use Business Intelligence Design Studio to create SQL Server Integration Services packages.

Estimated lesson time: 60 minutes

Business Intelligence Development Studio

BIDS is a version of Microsoft Visual Studio 2005 that is focused on project types that are unique to SQL Server 2005 business intelligence. BIDS is not restricted to working on SSIS packages; it is also the primary environment you use to develop Analysis Services and Reporting Services projects.

BIDS projects are stored within solutions. Although you can create a solution first and then add projects to that solution, BIDS automatically creates a solution for you when you initiate a project.

Because this chapter focuses on SSIS, all projects in this chapter are based on the Integration Services Project template. To initiate a project in which you create an SSIS package, first launch BIDS by choosing SQL Server Business Intelligence Development Studio from the Microsoft SQL Server 2005 program menu. From the File menu, choose New and then Project. The New Project dialog box opens. From the Visual Studio Installed Templates pane of this dialog box, select Integration Services Project. Type a name for the project and then click OK.

After you have started an Integration Services Project, you are presented with the interface shown in Figure 7-1.

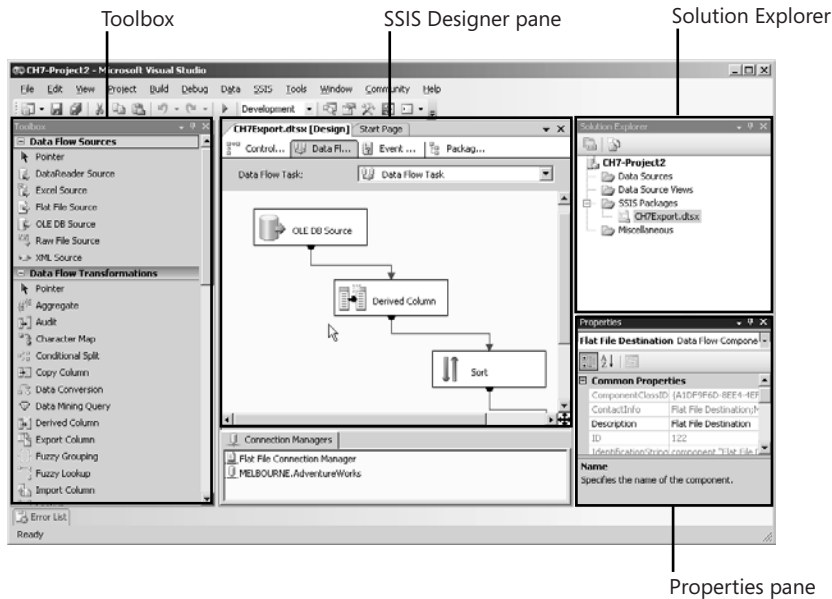


Figure 7-1 The BIDS interface.

BIDS consists of the following four main panes:

- **Toolbox pane** Located in the leftmost part of the window, this pane contains items that you can use in constructing projects. The availability of items is dependent on the current task.
- **SSIS Designer pane** Located in the center of the window, this pane is where you create or modify business intelligence objects.
- **Solution Explorer pane** Located in the top right of the window, this pane contains all items associated with the current project.
- **Properties pane** Located in the bottom right of the window, this pane contains the properties of an object.

You use each of these panes continually when constructing complex SSIS packages. In the next section, the function of each pane is explored in more detail. When you have an understanding of the functionality of each pane, move on to the discussion of how you can use the panes to meet the exam objective of constructing complex SSIS packages.

Toolbox Pane

At its most basic, the Toolbox is a collection of tasks that varies depending on what you are doing in the rest of BIDS. You drag tasks from the Toolbox to the Designer pane as you build your project. The Toolbox contains sections, which are collections of tasks that are relevant to what you are currently doing—for example:

- When you are editing an SSIS package's control flow, the Control Flow Items and Maintenance Plan Tasks sections are the only sections available in the Toolbox.
- When you are editing an SSIS package's Data Flow, the Data Flow Sources, Data Flow Transformations, and Data Flow Destinations sections are the only sections available in the Toolbox.

There are more than 40 tasks available in the Toolbox when the Control Flow pane is in focus and 50 different ones available in the Toolbox when the Data Flow pane is in focus. Given the sheer number of tasks, rather than trying to memorize the functionality of each one, you should open BIDS, hold the mouse pointer over each item, and read the balloon description that appears. Later on, after you have a basic understanding of how to put packages together, you should experiment. Most people learn better by applying knowledge practically rather than just absorbing it from a page.

SSIS Designer Pane

SSIS Designer is a component of the Business Intelligence Development Studio, and you can use it to perform the following tasks:

- Construct the control flow in a package.
- Construct the data flow in a package.
- Add event handlers to the package and package objects.
- View package content.
- View the execution progress of a package.

SSIS Designer has four permanent panes. The first is used for building package control flow, the second is for data flow, the third is for event handlers, and the final one is for viewing package contents. When a package is executing, a fifth tab appears that displays progress and execution results.

Solution Explorer Pane

In the Solution Explorer pane, you can create empty solutions and then add new or existing projects to those solutions. From Solution Explorer, you can perform management tasks such as Building, Rebuilding, and Debugging projects; execute

packages; and add miscellaneous files to a project. When working with Integration Services packages, it is also possible to configure new Data Sources, Data Source Views, and SSIS packages for inclusion within a project by using the Solution Explorer pane.

Properties Pane

The Properties pane is where you configure items on the package and task container levels. When you select an item in the Control Flow, Data Flow, or Solution Explorer panes, the Properties pane displays the appropriate configuration settings for that item. For example, if you select a particular task on the Control Flow pane, you can configure the property that determines whether that particular task can be used as a restart point in the Properties pane. By clicking in the background of the Control Flow pane when a package is selected in the Solution Explorer, you can configure security settings for the package, such as whether sensitive information will be protected using a password. The content of the Properties pane is discussed throughout this chapter.

MORE INFO BIDS

You can find out more about the different aspects of BIDS by accessing the MSDN article "Introducing Business Intelligence Development Studio" at msdn2.microsoft.com/en-us/library/ms173767.aspx.

SSIS Packages

At its most basic level, a *package* is an organized collection of tasks and workflow elements. The order of task execution is dependent on the outcome of earlier steps in the workflow. Depending on the result of a task's execution, a branch might occur. For example, if task alpha successfully executes, task beta executes next; however, if task alpha does not successfully execute, task gamma executes next.

You can construct packages visually by using BIDS or programmatically by using an IDE such as Visual Studio. In this chapter, you use BIDS only to construct and manage several different packages. This coverage is limited in part because the 70-444 exam is aimed at database administrators rather than at database developers and, as an administrator, you do not likely need to submerge yourself in source code.

When you finish creating packages, you can save them to a SQL Server's msdb database or to an XML structured .DTSX file. Each approach has benefits and drawbacks, and these are explored later in the chapter.

Packages include one or more of the following components:

- Checkpoints and restarts
- Configurations
- Connection managers
- Control flow elements
- Data flow elements
- Data tasks
- Event handlers
- Logging
- Security settings
- SSIS variables
- Transaction attributes

Although all these terms might seem a little intimidating, they are explained fully throughout the chapter with examples. These terms are also explained in the glossary at the end of the book.

Understanding the Differences Between Control Flow and Data Flow

The process of creating SSIS packages involves collecting together control flow and data flow tasks. Although it is possible to create an SSIS package without any data flow tasks, most SSIS packages consist of a combination of the two types.

At its simplest, you can think of a *control flow task* as a task that generally does something with the files that contain data, such as copying a file or executing a script. A *data flow task* does something with the data itself—for example, extracting data from a database table and sorting it. In combining the two, you might decide to extract data, make changes to it, and then use a control flow task to copy the result somewhere else.

To include data flow tasks in your SSIS package, you must drag a data flow task onto the Control Flow pane and select it. When the data flow task is selected, you need to navigate to the Data Flow pane to configure the data flow for that task. If you have multiple data flow tasks on the Control Flow pane, the information displayed on the Data Flow pane will change depending on which data flow task is selected on the Control Flow pane.

Quick Check

1. Which pane in BIDS would you drag a File System task onto?
2. What do you need to do if you want to include a data flow task in your package?
3. Where can you save packages in BIDS?

Quick Check Answer

1. You would drag a File System task onto the Control Flow pane.
2. Drag a data flow task to the Control Flow pane.
3. You can save packages in BIDS in the file system or MSDB database.

Differences Between DTS and SSIS Packages

SSIS is the replacement subsystem for SQL Server 2000 Data Transformation Services (DTS). Backwards compatibility for DTS packages is provided to help organizations with the transition between SQL Server 2000 and SQL Server 2005. When you install SSIS, the following are also installed: DTS runtime, package enumeration, and the Package Migration Wizard. Whereas DTS packages are COM-based, SSIS packages are Microsoft .NET Framework–compliant and support .NET languages. No DTS package editor is provided with SSIS, so if you want to make changes to your DTS package, you'll need to migrate it to SSIS.

MORE INFO Package Migration Wizard

For more information about the upgrading DTS packages, see the article at: msdn2.microsoft.com/en-us/library/ms143496.aspx.

Building a Package

There are two different approaches to building a package. In the first approach this section describes, you drag tasks from the toolbar onto either the Control Flow or Data Flow pane and link the tasks together. In the second approach, you follow the SQL Server Import And Export Wizard to build a basic package.

Using Control Flow or Data Flow

Building a package involves dragging tasks from the toolbar onto either the Control Flow or Data Flow panes and linking them together. You can link two or more tasks on the Control Flow pane by selecting the first one and dragging an arrow that

appears underneath to the second task. You can have multiple links from one task to another. After you link two tasks, you can right-click the link and specify whether it is activated when the previous task has completed successfully, when the previous task has failed, or when the previous task has executed regardless of the previous task's success or failure, as shown in Figure 7-2. The nature of the links between tasks is visually represented by three different line colors:

- **Green** The linked task will execute only if the prior task completes successfully.
- **Blue** The linked task will execute if the prior task completes. It does not matter whether the prior task completes successfully or completes with a failure.
- **Red** The linked task will execute if the prior task fails.

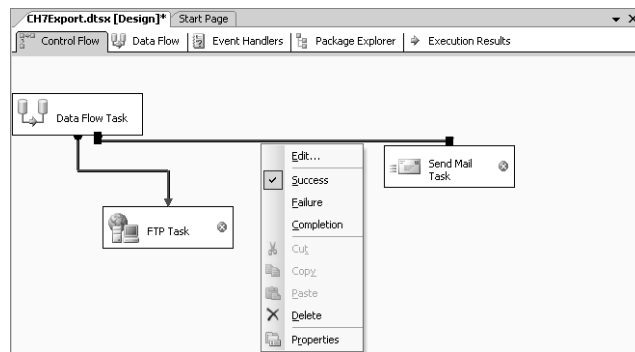


Figure 7-2 Constructing paths between tasks.

Double-clicking an item that you've dragged across to the Control Flow or Data Flow pane allows you to edit its properties. The properties of each item are unique to that item. For example, the properties of the FTP task in Figure 7-2 involve configuring destination servers and whether files are downloaded or uploaded; the properties of the Send Mail task involve configuring properties of the e-mail that will be sent. Because it isn't feasible to go through each task in the Toolbox and describe how you configure it without making this section five times longer, you should explore each task yourself by dragging the tasks across to the Control Flow or Data Flow pane and then viewing their properties.

Exam Tip Developing SSIS packages can be daunting to systems administrators who got into administration because they didn't like to write code. When you come across a question that asks you to build an SSIS package, make sure that you don't panic! Take a deep breath and break the question down in your mind. Think about what tools the exam has presented you with and how you could use these tools to solve the problem.

How you build an SSIS package depends on what you want the package to do. The best place to start is to break down the package that you want to build into separate tasks and then join those tasks together on the Control Flow pane. Breaking a problem into small pieces can simplify it greatly.

MORE INFO Designing and creating Integration Services packages

There is a wealth of information on designing and creating SSIS packages, including more detail on the functionality of Toolbox items, located on MSDN at [msdn2.microsoft.com/en-us/library/ms141091\(sql.90\).aspx](http://msdn2.microsoft.com/en-us/library/ms141091(sql.90).aspx).

SQL Server Import And Export Wizard

Another way to create basic packages is to use the SQL Server Import And Export Wizard. The wizard allows you to create a basic SSIS package that copies data to and from the following sources:

- SQL Server
- Flat files
- Microsoft Access
- Microsoft Excel
- OLE DB providers
- Microsoft .NET Data Provider for mySAP Business Suite.

You can use ADO.NET providers as sources for the wizard, but you cannot use them as destinations.

You can start the SQL Server Import And Export Wizard either from SSMS, from the command line, or from BIDS. To start the wizard from SSMS, connect to the database server, right-click a database, choose Tasks, and then choose either Import Data or Export Data. It is also possible to start the SQL Server Import And Export Wizard from the command line using the DTSWizard.exe command, located in `c:\Program Files\Microsoft SQL Server\90\DTS\Binn`. The Wizard is shown in Figure 7-3.

To start the wizard from BIDS, start a new project and do one of the following tasks:

- Right-click the SSIS Packages folder and choose SSIS Import And Export Wizard.
- Select SSIS Import And Export Wizard from the Project menu.

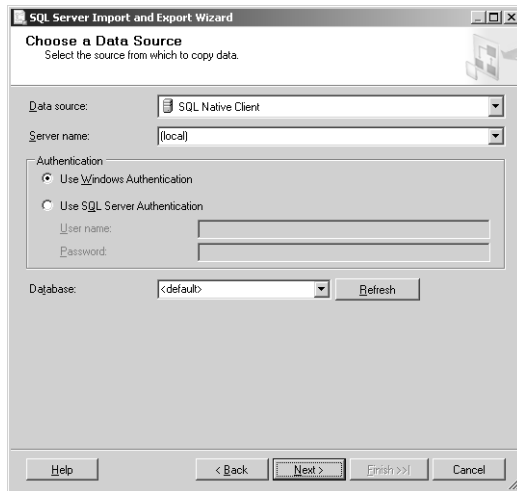


Figure 7-3 SQL Server Import And Export Wizard.

The SQL Server Import And Export Wizard is designed to create only basic SSIS packages. To accomplish more advanced tasks, you need to use the full functionality of the SSIS Designer in BIDS. The SQL Server Import And Export Wizard does not support column-level transformations and provides little in the way of transformation capabilities other than setting names, data type, and data type properties in destination files and tables. It is possible to edit existing packages that you created with the SQL Server Import And Export Wizard by using the SSIS Designer.

Executing Packages

You can use the Execute Package Utility and the dtexec command-line utility to run packages in development and production environments. You use both tools for instant rather than scheduled package execution. If you are going to use the Execute Package Utility, you must ensure that the Integration Services service is running. Scheduling package execution is covered in more detail in Lesson 4.

To execute packages using the Execute Package Utility, perform the following steps:

1. Open SSMS on the SQL Server 2005 computer on which the packages are installed, and click Connect in Object Explorer to establish a connection to the local Integration Services.
2. Locate the Stored Packages folder under Integration Services. Depending on how you have installed the packages, they are located under either the File System or the MSDB node.

- Right-click the package that you want to execute and choose Run Package. This initiates the Execute Package Utility shown in Figure 7-4.

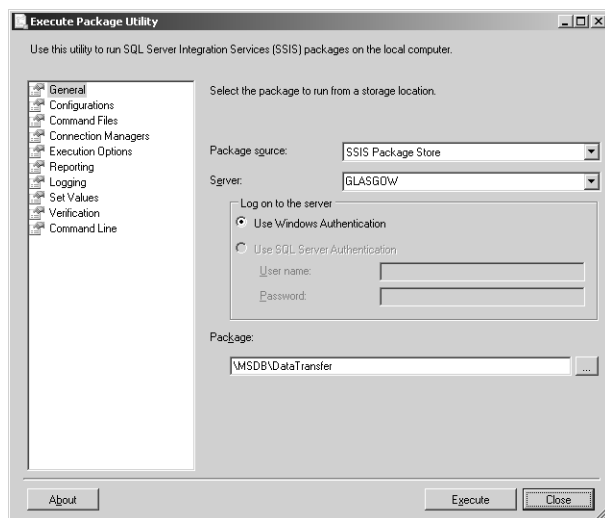


Figure 7-4 The Execute Package Utility.

- Click Execute and the package executes, displaying the Package Execution Progress dialog box so that you can view in-progress execution results. If necessary, you can click Stop to halt execution.

PRACTICE Creating an SSIS Package

In these practices, you create both a basic and a more advanced SSIS package using the SSIS Designer in Business Intelligence Development Studio.

► Practice 1: Creating a Simple Package

In this exercise, you create a basic package using SQL Server Business Intelligence Development Studio that uses Notepad to create a text file and then copies the created file to a separate directory.

- Create the directories `c:\temp1` and `c:\temp2` on the SQL Server 2005 computer.
- Open the SQL Server Business Intelligence Development Studio.
- From the File menu, choose New and then Project.
- Select the Integration Services Project from the list of Visual Studio installed templates.
- Give the project the name `CH7-Project1`, and ensure that the Create Directory For Solution check box is selected, as shown in Figure 7-5. Click OK.

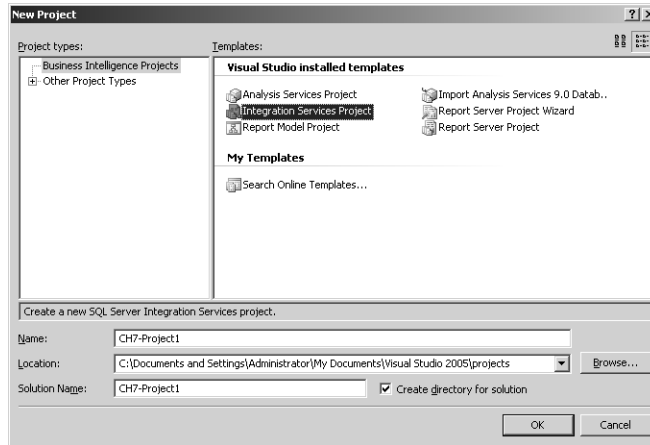


Figure 7-5 Creating a new project.

6. From the Control Flow Items pane of the Toolbox, drag the Execute Process task across to the center of the Design pane. Double-click the Execute Process task to open the Execute Process Task Editor dialog box.
7. On the General tab in the Name text box, name the task **Notepad**. Click the Process tab, and then type **notepad.exe** in the Executable text box.
8. In the Arguments text box, type **c:\temp1\example.txt**, as shown in Figure 7-6, and then click OK to close the Execute Process Task Editor dialog box.

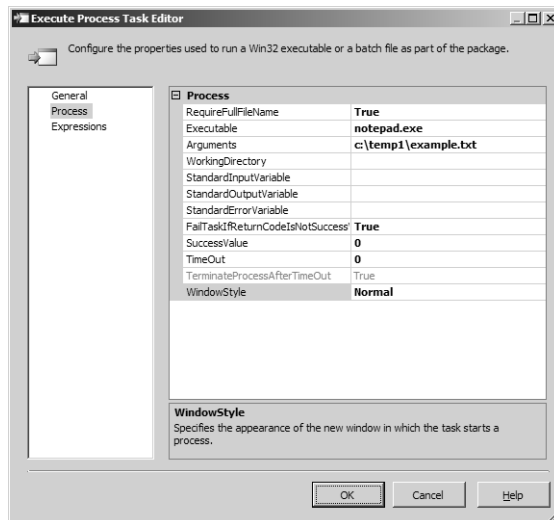


Figure 7-6 Configuring the Notepad task.

9. From the Control Flow Items section of the Toolbox, drag the File System Task across to the Design pane to a spot near the existing Execute Process task.
10. Double-click the File System Task to configure it.
11. In the Name text box, type **Copy Files**. In the DestinationConnection drop-down list, select <New connection...>. This opens the File Connection Manager Editor dialog box.
12. In the Usage Type drop-down list, select Existing Folder.
13. Click the Browse button, and select the c:\temp2 folder. Click OK to close the Browse For Folder dialog box.
14. Click OK to close the File Connection Manager Editor dialog box.
15. Ensure that the Operation is set to Copy File. Set IsSourcePathVariable to True. In the SourceVariable drop-down list, select <New variable...>. This opens the Add Variable dialog box.
16. Change the Container to File System Task, set the name of the variable to **Text-File**, and set the Value text box to **c:\temp1\example.txt** as shown in Figure 7-7. Click OK twice to return to the Design pane.

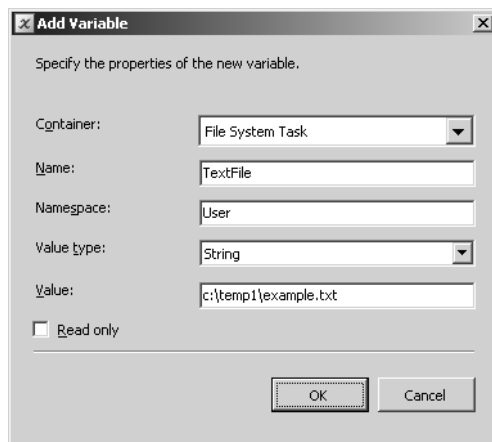


Figure 7-7 The Add Variable dialog box.

17. In the Design pane, select the Execute Process Notepad task. Drag the green arrow that appears under the task so that it connects with the File System task.
18. From the Debug menu, choose Start Debugging.

19. Notepad is launched, and you are asked whether you want to create a new file. Click Yes. Type **I will pass the 70-444 exam** into Notepad and then save and close the file.
20. From the Debug menu, choose Stop Debugging. Close BIDS, and verify the existence of the example.txt file in the c:\temp2 directory.

► **Practice 2: Creating a Data Flow Package**

In this practice, you create a more complicated data flow package that retrieves data from the sample AdventureWorks database, makes alterations to it, and then writes it to a flat file.

1. Open the SQL Server Business Intelligence Development Studio.
2. From the File menu, choose New and then Project.
3. Verify that the Integration Services Project template is selected from the list of Visual Studio installed templates.
4. Give the project the name CH7-Project2, ensure that the Create Directory For Solution check box is selected, and click OK.
5. In the Solution Explorer pane, right-click Package.dtsx and rename it to **CH7Export.dtsx**. Click Yes to rename the package object.
6. Drag a Data Flow Task from the Toolbox onto the Control Flow pane, and then double-click the Data Flow Task to open the Data Flow pane.
7. Drag an OLE DB Source onto the Data Flow pane from the Data Flow Sources section of the Toolbox.
8. Double-click the OLE DB Source to open the OLE DB Source Editor dialog box.
9. Click New next to OLE DB Connection Manager, and then click New again to open the Connection Manager dialog box.
10. Set Server Name to the SQL Server 2005 computer that you are using.
11. In the Connect To A Database drop-down list, select the AdventureWorks database.
12. Click OK twice to return to the OLE DB Source Editor dialog box.
13. In the Data Access Mode drop-down list, select Table Or View, and use the Name Of The Table Or The View drop-down list to select the [HumanResources].[EmployeePayHistory] table, as shown in Figure 7-8.

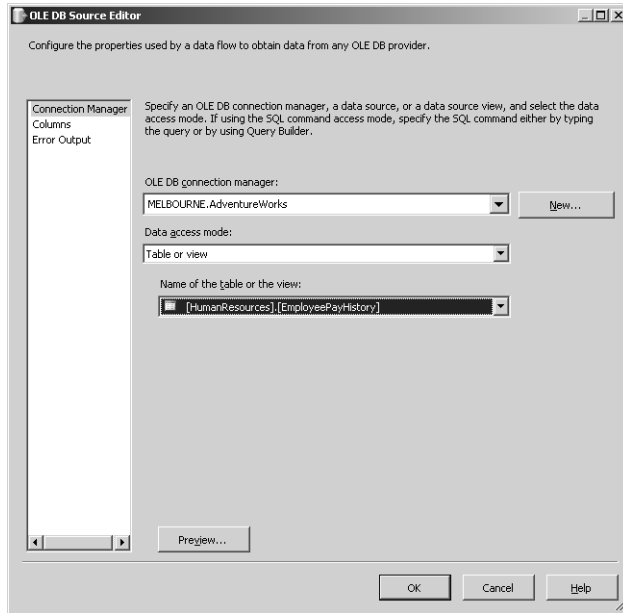


Figure 7-8 OLE DB Source Editor.

14. From the list on the left side of the page, choose Columns. Ensure that only the EmployeeID, Rate, and PayFrequency check boxes are selected. Click OK to close the OLE DB Source Editor dialog box.
15. Drag the Derived Column task to a spot under the OLE DB Source task from the Data Flow Transformations section of the Toolbox.
16. Select the OLE DB source, and drag the green arrow that appears underneath to the Derived Column task.
17. Double-click the Derived Column task to open the Derived Column Transformation Editor dialog box.
18. In the Derived Column Name text box, type **TotalSalary**.
19. Expand Columns in the top left box, and drag both Rate and PayFrequency into the Expression field. Type an asterisk (*) between [Rate] and [PayFrequency] as shown in Figure 7-9. Click OK to close the Derived Column Transformation Editor dialog box.
20. Drag a Sort task from the Data Flow Transformations section of the Toolbox to a section of the Data Flow pane under the Derived Column task. Select the Derived Column task. Drag the green arrow that appears onto the Sort task.

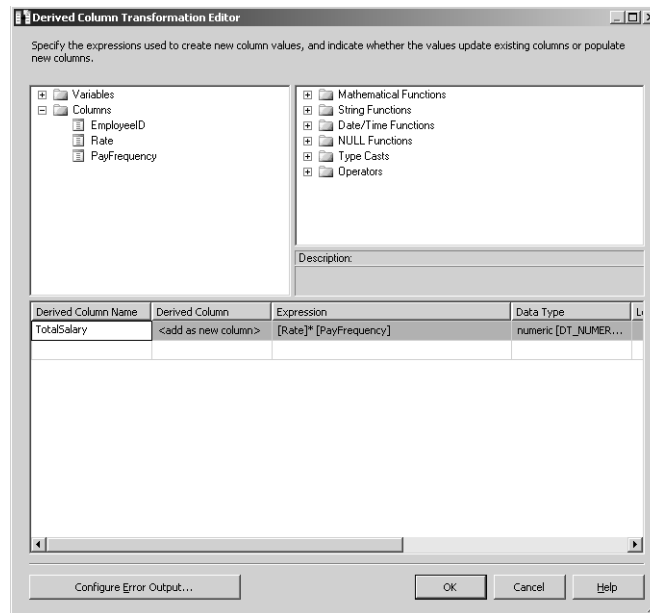


Figure 7-9 The Derived Column Transformation Editor.

21. Double-click the Sort task to open the Sort Transformation Editor dialog box. Select the TotalSalary column check box, and change the Sort Type to descending. Click OK to close the Sort Transformation Editor dialog box.
22. Drag a Flat File Destination task from the Data Flow Destinations section of the Toolbox to a place on the Data Flow pane under the Sort task. Select the Sort task, and drag the green arrow that appears onto the Flat File Destination task.
23. Double-click the Flat File Destination task to open the Flat File Destination Editor dialog box. Click New to open the Flat File Format dialog box. Verify that the Delimited format is selected for the destination flat file, and click OK. In the File Name text box, type `c:\temp1\output.csv`. Select the Column Names In The First Data Row check box. Click OK. From the list on the left side, select Mappings and then click OK to close the Flat File Destination Editor dialog box.
24. From the Debug menu, choose Start Debugging. Ensure that each task completes successfully. Open the file `c:\temp1\output.csv` using Notepad, and verify that the output is sorted by TotalSalary from highest to lowest.
25. From the Debug menu, choose Stop Debugging. Save the project and close BIDS.

Lesson Summary

- A package is an organized collection of tasks and workflow elements.
- The order of task execution is dependent on the outcome of earlier steps in the workflow.
- Packages can be constructed visually using BIDS.
- Packages can be saved to a SQL Server's msdb database or to an XML structured .DTSX file.
- The process of creating SSIS packages involves collecting together control flow and data flow tasks.
- To include data flow tasks in your SSIS package, drag a Data Flow Task onto the control flow, select it, and then navigate to the Data Flow pane to configure the task.
- Double-clicking an item that you've dragged across to the Control Flow or Data Flow pane allows you to edit its properties.
- The Execute Package Utility and the dtexec command-line utility can be used to run packages in development and production environments.

Lesson Review

You can use the following questions to test your knowledge of the information in Lesson 1, "Constructing SSIS Packages." The questions are also available on the companion CD if you prefer to review them in electronic form.

NOTE Answers

Answers to these questions and explanations of why each answer choice is correct or incorrect are located in the "Answers" section at the end of the book.

- I. You want to transform data located in the AdventureWorks database; which of the following tasks would you drag to the Control Flow pane?
 - A. Data flow task
 - B. File System task
 - C. FTP task
 - D. Send Mail task

2. You are examining an SSIS package designed by your assistant. A red line connects a File System task to an FTP task. Under which conditions will the FTP task execute?
 - A. If the File System task succeeds
 - B. If the File System task completes
 - C. If the File System task fails

Lesson 2: Securing SSIS Packages

SSIS packages can contain sensitive information such as passwords or a connection string. Packages can also make significant changes to a database or be used to extract sensitive information. For these reasons, it is not only important to secure the operation of the package, but to ensure that only authorized persons have access to the sensitive data located inside packages.

After this lesson, you will be able to:

- Design security for accessing SSIS packages.

Estimated lesson time: 30 minutes

Securing Sensitive Data Using Package Protection Levels

When you design a package by using BIDS, you sometimes need to add sensitive information so that the package can properly interact with SQL Server. SSIS automatically detects sensitive properties such as connection strings and deals with them according to the assigned package protection level. For example, if you set a package to a protection level that encrypts sensitive information by using a password, SQL Server encrypts the values of every property it identifies as sensitive. When writing your own custom tasks, connection managers, or data flow components, it is possible to specify which properties Integration Services should treat as sensitive.

When configuring package protection levels, you can encrypt just the sensitive information or the entire package. Encryption is based either on a supplied password or a user key.

The different package protection levels are described in Table 7-1.

Table 7-1 Package Protection Levels

Protection Level	Description
DontSaveSensitive	Properties marked as sensitive are not saved with the package. When another user opens the package, SQL Server replaces sensitive information with blanks. It then requires the user who opens the package to provide the sensitive data.
EncryptAllWith-Password	Encrypts the entire package by using a password that the user provides when the package is created or exported. Users must provide the package password to open the package in BIDS or to run it using the dtexec utility.

Table 7-1 Package Protection Levels

Protection Level	Description
EncryptAllWithUserKey	Encrypts the entire contents of package based on the user key or the user who created or exported the package. Only the user who created or exported the package can open the package in BIDS or run it using the dtexec utility.
EncryptSensitive-WithPassword	Encrypts information marked as sensitive using a password. Sensitive data is saved as a part of the package and encrypted with a user-supplied password supplied during package creation or export. If a user attempts to open the package in BIDS without supplying the password, he or she must provide new values for sensitive data. Package execution fails if a user attempts to execute the package without providing the password.
EncryptSensitive-WithUserKey	Encrypts sensitive information using the current user key. If a different user opens the package in BIDS, SQL Server replaces the sensitive information by blanks and the current user must enter new values. If a user other than the one who created the package attempts to execute it, execution fails.
ServerStorage	The entire package is protected using SQL Server database roles. Only supported if you save the package to the SQL Server msdb database. It is not supported if you save the package to the file system from BIDS.

To set the package protection level on a package, navigate to the Package Explorer tab in the main BIDS window; in the Properties window, scroll down to the Security section as shown in Figure 7-10. Select the value next to ProtectionLevel to change it to one of the values listed in Table 7-1.

Database-Level Role Security

It is also possible to secure SSIS packages by saving them within the msdb database and assigning one of the fixed database-level roles to the package. The three fixed database-level roles are db_dtsadmin, db_dtsltduser, and db_dtsoperator. It is important to note that roles can be implemented only on packages that are saved to the msdb database in SQL Server and not on packages stored within the file system. You assign

roles to a package using SSMS, and SQL Server saves these role assignments to the msdb database. Each database role has the read and write action properties listed in Table 7-2.

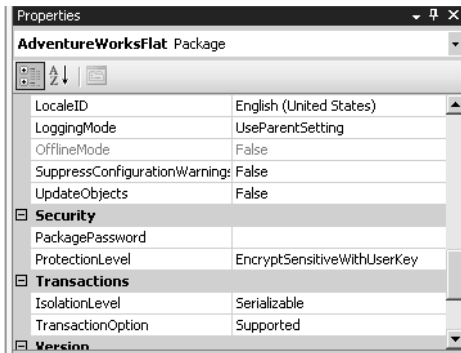


Figure 7-10 Configuring package protection.

Table 7-2 Fixed-Level Role Read and Write Actions

Role	Read Action	Write Action
db_dtsadmin	Enumerate all packages. View all packages. Execute all packages. Export all packages. Execute all packages in SQL Server Agent.	Import packages. Delete all packages. Change all package roles.
db_dtsltduser	Enumerate all packages. View own packages. Execute own packages. Export own packages.	Import packages. Delete own packages. Change own package roles.
db_dtsoperator	Enumerate all packages. View all packages. Execute all packages. Export all packages. Execute all packages in SQL Server Agent.	None.

Table 7-2 Fixed-Level Role Read and Write Actions

Role	Read Action	Write Action
Windows Administrators	View all running packages.	Stop all currently running packages.
Non-Windows administrators	View packages that they started.	Stop packages that they started.

Because Windows Administrators can view and stop packages running on a SQL Server computer, be sure to restrict access to SQL Server computers to only those users for whom you want to have this functionality. For example, you might create a separate domain in your organization's forest to ensure that users who have been assigned domain administrator privileges in one of your forest's domains don't automatically have administrative access to your SQL Server computers.

If you want to create your own user-defined roles and apply them to packages, you must first add these roles to the msdb database. Only when you added them to the msdb database can you apply them to packages.

You create new database roles in SSMS. To create a new database role, expand Databases\System Databases\msdb\Security and right-click the Roles node. Choose New and then New Database Role. Provide a name, owner, and owned schemas; and then add role members. Although it is also possible to edit and extend permissions through the application of user-defined roles to SSIS packages, this functionality is likely beyond the scope of the 70-444 exam.

To assign Reader and Writer roles to SSIS packages already stored within the msdb database, you should perform the following steps:

1. In Object Explorer within SSMS, click Connect, and select Integration Services. Click Connect to connect to the local Integration Services service.
2. After the connection is established, locate the Integration Services connection and expand the Stored Packages folder. Locate the subfolder containing the package to which you are going to assign roles.
3. Right-click the package and choose Package Roles. This launches the Package Roles dialog box shown in Figure 7-11.
4. Select a reader role in the Reader Role drop-down list and a writer role in the Writer Role drop-down list. Click OK to exit the dialog box and save your changes.

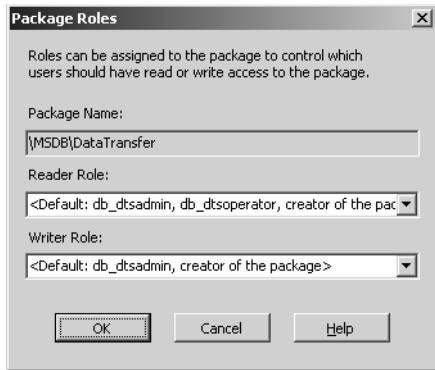


Figure 7-11 Package Roles dialog box.

Package and Configuration Storage Security

Although you generally store packages within the msdb database, you can also save packages to the file system as XML files using the .dtsx file name extension. If you choose to save packages to the file system in XML format, you should protect these files using NTFS file and folder permissions. The mechanics of saving, deploying, and exporting packages is covered in more detail in Lesson 4, “Deploying SSIS Packages.”

Unlike packages, which, if you choose to save them to the database, can be saved only to the msdb database, you can save package configurations to a specially configured table in any SQL Server 2005 database. If the table that will contain the package does not exist, Integration Services automatically creates it using the name you enter in the Save Configuration dialog box. Integration Services also ensures that the table has the correct structure. You can save multiple package configurations to the same table. Saving to a table provides security through permissions assigned at the server, database, and table levels.

It is also possible to save package configurations to the file system. You should also secure package configurations you save to the file system through the use of NTFS permissions. Packages you configure to use checkpoints and logging generate data that is stored externally to the package. Although logging data can be saved to SQL Server database tables, checkpoint files can be saved only to the file system.

Digitally Signing Packages

A final method of ensuring package security is by digitally signing packages. It is possible to sign an SSIS package with a digital certificate and require the runtime to check the signature’s validity before loading the package. Digitally signing packages

prevents packages that have been altered from loading and running. This means that if you alter a package that has already been digitally signed, you must re-sign the package when you complete your alterations. It is important to note that the certificate you use to sign the package must be issued by a trusted certificate authority and that you can sign packages only by using certificates created for the purpose of code signing. The *CheckSignatureOnLoad* and *CertificateObject* properties of the package determine whether the package's signature must be checked and the particular code signing certificate used to authorize the package.

Quick Check

1. Which security setting would you use if you wanted to protect sensitive data and allow your partner to edit a package without having to reenter sensitive information?
2. In which database can you store SSIS packages?

Quick Check Answer

1. `EncryptSensitiveWithPassword`
2. MSDB database

After you have obtained a code signing certificate from a trusted Certificate Authority, you can use the following steps to digitally sign SSIS packages:

1. Open the Integration Services project that contains the package you want to sign in BIDS. Double-click the package in Solution Explorer to open it.
2. From the SSIS menu, choose Digital Signing. In the Digital Signing dialog box, click Sign. Doing this opens the Select Certificate dialog box. Select the code signing certificate and click OK.
3. Click OK to close the Digital Signing dialog box. Save the updated package.

Code signing certificates do not have to be unique to a particular package. It is possible to sign many different packages using the same code signing certificate.

PRACTICE Securing an SSIS Package

In this practice, you encrypt the SSIS package that you created in the first practice of Lesson 1, "Constructing SSIS Packages," by using a password. You will then verify that this has been done correctly.

1. Using BIDS, open CH7-Project1.

- In the Properties pane, scroll down to Security. Change the ProtectionLevel field to EncryptAllWithPassword, as shown in Figure 7-12. In the PackagePassword field, click on the (...) button. This opens the Property Editor dialog box.

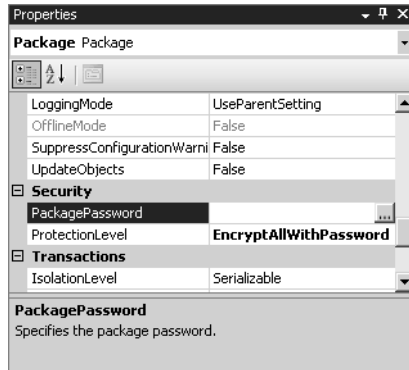


Figure 7-12 Configuring a package password.

- Type the password **P@ssw0rd** twice. Click OK.
- On the toolbar, click Save All and then exit BIDS.
- Restart BIDS and attempt to open CH7-Project1. You should now see the Package Password dialog box. Type the password you configured in step 3 to edit the package and then click OK.

Lesson Summary

- By editing the package protection level properties, you can encrypt either just the sensitive information or the entire package. Encryption is based on the user's key or a password.
- You can store packages, package configurations, and log files within specific SQL Server databases. These packages are secured using appropriate database and table-level security.
- You can use three built-in database roles—`db_dtsadmin`, `db_dtsltduser`, and `db_dtsoperator`—to secure packages stored within SQL Server's `msdb` database.
- You should use NTFS permissions to protect packages, package configurations, log files, and checkpoint information stored within the file system.
- You can stop a modified package from executing by digitally signing the package with a code-signing certificate.

Lesson Review

You can use the following questions to test your knowledge of the information in Lesson 2, “Securing SSIS Packages.” The questions are also available on the companion CD if you prefer to review them in electronic form.

NOTE Answers

Answers to these questions and explanations of why each answer choice is correct or incorrect are located in the “Answers” section at the end of the book.

1. Which of the following SSIS package elements can be saved only to the file system and cannot be saved to a SQL Server database?
 - A. SSIS packages
 - B. SSIS package configurations
 - C. SSIS package logs
 - D. SSIS package checkpoint files
2. Ian and Orin are developing SSIS packages together. They are storing the package files on a file server’s shared folder. Ian and Orin have agreed on a shared password that they will use to protect sensitive information stored within packages. When Ian creates a package and configures the package protection level, which of the following settings should he not use to ensure that Orin doesn’t have to re-enter sensitive values when he opens the package? (Choose all that apply.)
 - A. EncryptSensitiveWithUserKey
 - B. EncryptSensitiveWithPassword
 - C. EncryptAllWithUserKey
 - D. EncryptAllWithPassword
3. What type of package security would you use to ensure that the contents of a package have not been altered since it was created?
 - A. Encrypting the entire package by using the user’s key
 - B. Storing the package within the SQL Server’s msdb database
 - C. Encrypting the package with a password
 - D. Digitally signing the package

4. Which of the following built-in roles, when appropriately assigned, are able to export all packages? (Choose all that apply.)
- A. db_dtsadmin
 - B. db_dtsltduser
 - C. db_dtsoperator

Lesson 3: Troubleshooting SSIS Packages

As any developer will tell you, rarely does any program work entirely correctly the first time it is executed. The same logic applies to SSIS packages. Only if you are extremely lucky does everything work the way you envision it when you first execute an SSIS package. In light of this reality, BIDS ships with many tools that you can use to debug your SSIS packages prior to deploying them in a production environment. These tools range from checkpoints, which allow you to restart a package from its point of failure rather than re-running the entire thing, to extensive logging options and breakpoints.

After this lesson, you will be able to:

- Restart failed packages.
- Troubleshoot or debug packages.

Estimated lesson time: 30 minutes

Package Checkpoints

Package checkpoints are a technology that allows packages to be restarted from the point of failure rather than re-executing the entire package. This feature can save a lot of time and effort, especially if the package fails during a minor task at the end of an exhaustive execution. The following is a list of examples that show the benefits of implementing checkpoints in SSIS packages:

- If a package uploads 10 large files by using the FTP task and the ninth file fails to upload, checkpoints enable that single file to be uploaded when the package is restarted rather than having to re-upload all 10 large files.
- If a package uses 20 different Bulk Insert tasks into different dimension tables and one of the Bulk Inserts fails, the use of checkpoints would mean that when the package was restarted only the failed task, and not the 19 others, will be retried.
- If a package computes 30 separate aggregates using 30 separate Data Flow tasks and one aggregation fails, restarting the package forces only the failed aggregation to be performed again rather than all 30.

Task host containers, Foreach Loop containers, and transacted containers are the smallest part of the work flow that can be restarted using checkpoints. If you want to use checkpoints, you should limit the number of tasks that you put in these types of containers as much as possible. The more granular you make your checkpoints, the more effective they are in stopping the repetition of unproblematic tasks.

Note that if a package fails during the execution of a transacted container, SQL Server rolls back any alterations made by the tasks in that container. When the package restarts, SQL Server reruns the transacted container from the beginning. Because a transacted container is the smallest part of the work flow that can be restarted, any of the transacted container's child containers that successfully executed prior to failure are rolled back and have to run again when the package is restarted. This has important implications for the integrity of data. For example, in cases where a series of changes are made, having half the tasks execute might be worse than having none of them execute.

This same data integrity issue applies to the child containers of Foreach Loop containers. When SQL Server restarts the package, the entire Foreach Loop container runs again. However, because this is not a transacted container, any changes made by child containers of the Foreach Loop containers that successfully ran prior to the package failing will be retained. Foreach Loop containers do not allow the rollback of changes if some aspect of the Foreach Loop container's execution fails.

Checkpoint files are used to monitor and manage the execution of packages that contain checkpoints. The checkpoint file is created when the package starts to execute, and it is removed after successful package execution. Checkpoint files are similar in many ways to bookmarks that SSIS can use to restart a package from its point of failure. Checkpoint files record the following information:

- The type of container that failed
- Current values of variables (other than those that are of the Object data type)
- Whether the failure occurred part way through the execution of a transaction

You should note that package configurations are ignored when packages are restarted because all the necessary information is drawn from the checkpoint file. Also note that packages can be restarted only at control flow levels, not at data flow levels. This will affect package design because you will want to minimize the number of data flows in any single data flow task.

To configure checkpoints, you need to configure the package properties fields listed in Table 7-3.

Table 7-3 Checkpoint Package Properties

Property	Value	Description
<i>CheckpointFileName</i>	Any	Specifies the name of the checkpoint file. You can use any file name.
<i>CheckpointUsage</i>	Never	Checkpoint files are not used.
	Always	The checkpoint file is used, and the package restarts from point of failure. If the file is not found, the package fails.
	IfExists	The checkpoint file is used if it exists. If the file is not found, the package restarts from beginning.
<i>SaveCheckpoints</i>	True/False	This property must be set to true to restart a package from point of failure.
<i>FailPackageOnFailure</i>	True/False	This property must be set to true for each container within the package that you want to identify as a restart point.

Incorporating Transactions into Packages

As discussed earlier, you can use transactions to allow a group of tasks in a package to be linked together as a single unit. If one of the collected tasks fails, the changes made by all other tasks can be rolled back. You can configure all SSIS container types to use transactions. The three options that SSIS includes for transaction configuration are as follows:

- **Required** The container starts a transaction unless one has already been started by a parent container. If a transaction was started by the parent container, the child container joins the existing transaction.
- **Supported** The container will not start a transaction, but it will join any transaction initiated by the parent container.
- **Nonsupported** The container will not start a transaction or join an existing transaction.

Figure 7-13 shows how you configure transactions by setting the *TransactionOption* property on each container.

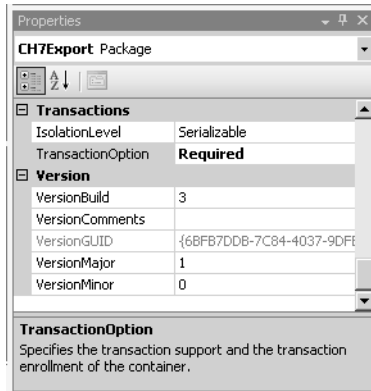


Figure 7-13 Configuring the *TransactionOption* property.

Package Debugging

When you execute a package using the debug menu, you can see whether each part of the package executes correctly. When a task is executing in SSIS Designer, the task icons change color depending on their status: yellow indicates the task is executing, grey indicates the task is waiting to execute, green indicates that the task has successfully executed, and red indicates that the task has failed to execute. This visual guide allows you to quickly locate problematic aspects of the control or data flow so that you can address them.

It is also possible to use the Progress tab, as shown in Figure 7-14, to identify errors in package execution. Errors appear in the Progress tab as red “x”s, which make the errors easy to identify.

Finally, the error list, located in the bottom left of the SSIS screen, provides basic descriptions of any errors or warnings that have occurred during package execution.

Data Viewers

Data viewers provide the ability for a developer to view data as it moves through a path. This feature allows the developer to identify bugs as they occur. You add data views by right-clicking the path between two data flow components and choosing the Data Viewer option. This opens the Configure Data Viewer dialog box. You can use this dialog box to add viewers that allow you to view data in grid, histogram, scatter plot, or column chart format, as shown in Figure 7-15.

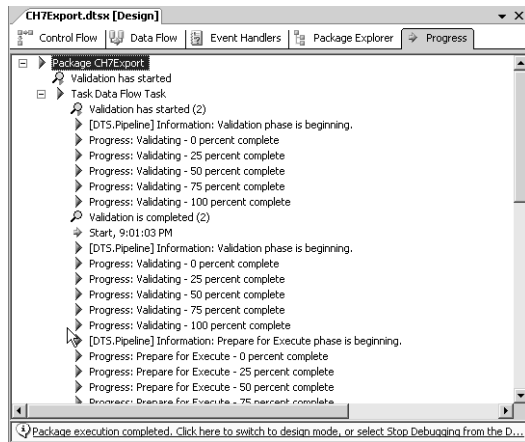


Figure 7-14 Viewing package execution in the Progress tab.

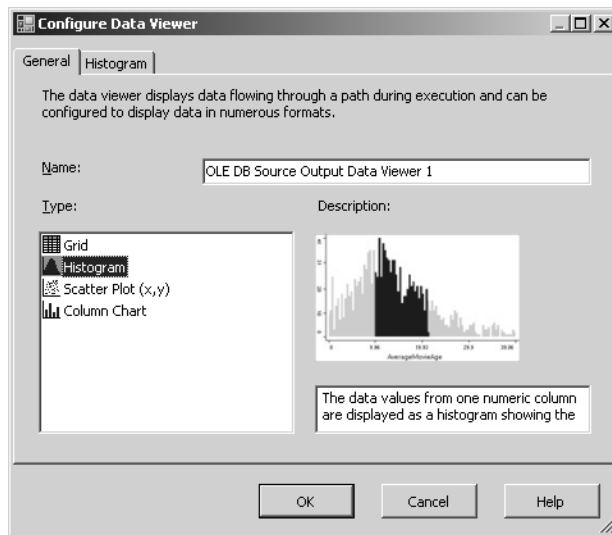


Figure 7-15 Configuring a data viewer.

Breakpoints Window

Using BIDS, you can set breakpoints within SSIS tasks, containers, and data-flow components. During package runtime, you can open the Breakpoints window, shown in Figure 7-16, which lists all breakpoints that you have enabled within the SSIS package. Breakpoints allow greater control over the debugging process. You access the Breakpoints window through the Debug menu. In the Breakpoints window, you can click New to add a new breakpoint to the package.

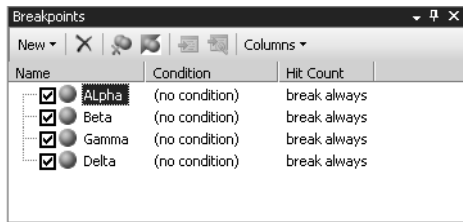


Figure 7-16 The Breakpoints window.

Package Logging

Logs are a powerful tool for troubleshooting SSIS packages. SSIS packages can write log entries when run-time events occur. You can also create custom log messages for use with SSIS packages. SSIS packages can write log entries to SQL Server Profiler, SQL Server, the Windows Event Log, or XML files. The method you choose is heavily dependent on the situation. For example, if you were to use Microsoft Operations Manager (MOM) to monitor whether packages have failed to execute, you would need to write log entries to the Windows Event Log provider. Logging can occur not only on the package level, but on the task and container level as well. You can enable logging for tasks and containers even though the package itself is not enabled for logging. And you can log different information from different tasks within the same container. Packages, tasks, and containers can also be configured to write events to multiple logs.

Quick Check

1. Which event log provider would you use if you wanted to have MOM monitor package execution?
2. Which technology allows greater control over the debugging process?
3. Which technology allows you to roll back a group of tasks?

Quick Check Answer

1. Windows Event Log
2. Breakpoints
3. Transactions

It is possible to configure a level of logging that suits your needs at a particular time. When you are developing a package, you might want to configure voluminous logs, but after the package has been shown to work in a production environment, you will likely want to reduce the number and type of events logged.

Real World

Orin Thomas

I've found that if I set my logs to be too sensitive, I miss important items in the sea of information. Tools such as Microsoft Log Parser, which can be found at www.microsoft.com/technet/scriptcenter/tools/logparser/default.aspx, can help you find the needle in the haystack, but the best approach is to keep the amount of information at a level where it doesn't become cumbersome.

To configure logging, perform the following steps:

1. Open the project that contains the package on which you want to configure logging.
2. If necessary, click the Control Flow tab, and then from the SSIS menu choose Logging. This opens the Configure SSIS Logs: Package dialog box.
3. In the Containers pane, select the check boxes for the items for which you want to generate log events. Then, select the specific item you want to log events from.
4. On the Providers And Logs tab, select a provider and click Add. The various provider options include XML files, text files, Windows Event Log, SQL Profiler, and SQL Server. It is possible to select more than one provider for each package, task, or container.
5. On the Details tab, select all events you want to log, as shown in Figure 7-17. Click OK to close the dialog box.

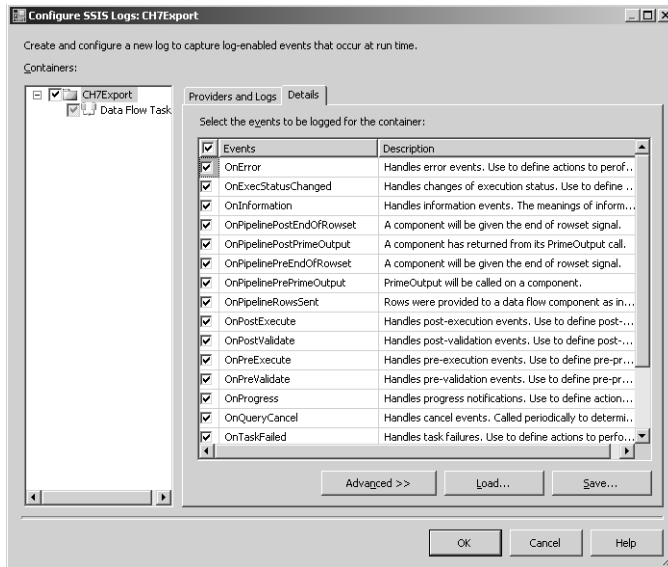


Figure 7-17 Configuring SSIS logs.

PRACTICE Setting Checkpoints

In the following practice, you configure an SSIS package to use checkpoints.

1. In BIDS, open an SSIS package.
2. In the Properties pane, navigate to the Checkpoints section as shown in Figure 7-18.

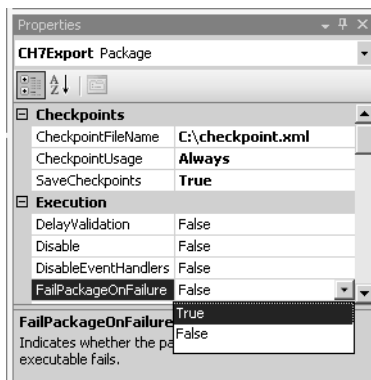


Figure 7-18 Configuring checkpoints.

3. Set the *CheckpointFileName* property to the file name you want to use for a checkpoint. There are no specific requirements for this file name.

4. Set the *CheckpointUsage* property to *IfExists*.
5. Set the *SaveCheckpoints* property to *True*.
6. Select each container in the Control Flow pane, one at a time. When you select a container, in the Properties pane set the *FailPackageOnFailure* property to *True*.

Lesson Summary

- The primary method for restarting failed packages is to institute checkpoints.
- Checkpoints allow task host containers, Foreach Loop containers, and transacted containers to be rerun from the point of failure rather than requiring you to restart the entire package.
- SSIS packages that are restarted using checkpoints pull variable data from the checkpoint file rather than from the package configuration.
- Transactions can be used to allow a group of tasks in a package to be linked as a single unit. If one of the tasks in the transactions fails, the alterations made by tasks that have already executed successfully can be rolled back and the group of tasks can be run again.
- You can visually debug tasks using the debugging menu. Tasks that are represented in yellow are executing, tasks shown in green have successfully executed, tasks shown in grey are waiting to execute, and tasks shown in red have failed to execute.
- The Progress tab provides a detailed runtime list of package execution elements.
- Data viewers can be added by right-clicking the path between two data flow components. The data viewers allow the developer to add viewers that display data transformations in real time.
- Breakpoints can be inserted in SSIS tasks, containers, and data-flow components, allowing greater control over the debugging process.
- Logging can be configured at the package, task, or container level. Logs can be written to SQL Server Profiler, SQL Server, the Windows Event Log, or XML files.

Lesson Review

You can use the following questions to test your knowledge of the information in Lesson 3, “Troubleshooting SSIS Packages.” The questions are also available on the companion CD if you prefer to review them in electronic form.

NOTE Answers

Answers to these questions and explanations of why each answer choice is correct or incorrect are located in the “Answers” section at the end of the book.

1. You have created an SSIS package that uses 10 FTP tasks to download files each evening from several servers on a remote network. Although you’ve set the *SaveCheckpoints* property to *True*, set the *CheckpointUsage* property to *Always*, and created a checkpoint file and set the *CheckpointFileName* property to point to this file name, when one of the FTP tasks fails and you restart the package, the package appears to restart from the beginning. How can you ensure that the package restarts from the point of failure?
 - A. Ensure that the *FailPackageOnFailure* property is set to *True* on each FTP task container.
 - B. Ensure that the *FailPackageOnFailure* property is set to *False* on each FTP task container.
 - C. Set the package’s *SaveCheckpoints* property to *False*.
 - D. Set the package’s *CheckpointUsage* property to *IfExists*.
2. The *TransactionOption* property on a parent container is set to *Supported*, and the *TransactionOption* on the child container is set to *Required*. Which of the following statements are true? (Choose all that apply.)
 - A. Both the parent and child containers will be a part of the same transaction.
 - B. The parent container, but not the child container, will be a part of the same transaction.
 - C. The parent container will not be a part of a transaction.
 - D. The child container will be a part of a transaction.
3. Which of the following can you enable for logging if the SSIS package itself is not enabled for logging? (Choose all that apply.)
 - A. Foreach Loop containers
 - B. Control flow tasks
 - C. Data flow tasks
 - D. File System tasks

Lesson 4: Deploying SSIS Packages

After you have created your SSIS package and then debugged it, it is time to deploy the packages to the computers that host your organization's production database. The first step in deploying a package is to create a package configuration. The next step is the creation of the deployment utility. After you build the deployment utility, you can copy the collection of files to your target servers. To install the package on the target servers, you need to run the Package Installation Wizard. When you have verified that your SSIS package has installed correctly, it is time to configure a schedule for the package's execution using the SQL Server Agent. This final lesson covers all of these aspects.

After this lesson, you will be able to:

- Deploy and move packages.
- Schedule package execution.
- Move packages to different servers.

Estimated lesson time: 90 minutes

Package Configurations

Package configurations are the method that SSIS uses to update the values of properties within packages at run time. The advantage of using package configurations is that they do the following:

- Make deployment from development to production environments simpler by updating the package's connection manager connection string.
- Allow you to deploy packages intelligently. For example, they configure a check so that if the necessary amount of disk space a package needs to execute is not present on the target server that the package has been deployed to, the package will not execute.
- Allow packages to be more flexible by updating the value of variables that are used in property expressions.

There are several ways to store package configurations. Storage locations are listed in Table 7-4.

Table 7-4 Different SSIS Package Configuration Locations

Location	Properties
XML configuration file	A single file can store one or many package configurations.
Environment variable	Can store a single package configuration.
Registry entry	Can store a single package configuration.
Parent package configuration	Package variable stores configuration. Used to update properties in child packages.
SQL Server table	A table in the database holds one or many package configurations.

XML Package Configurations

XML configuration files allow a great degree of flexibility because you can create a new XML configuration file, add a package configuration to an existing XML configuration file, or modify an existing XML configuration file using a new package configuration.

Registry Package Configurations

You can also store package configurations in a computer's registry. Keys are stored within the HKEY_CURRENT_USER hive. Keys are almost always directly off the root of HKEY_CURRENT_USER, though it is possible to specify an alternate location. You can use any name as long as the registry key itself has a value named *Value* that is a DWORD or a string. When selecting the Registry entry configuration type, enter the registry key name in the Registry Entry text box in the <registry key> format. For example, if you created a registry key called SSISPackageConfig off HKEY_CURRENT_USER, you would enter <SSISPackageConfig> in the Registry Entry text box of the Select Configuration Type page, as shown in Figure 7-19.

SQL Server Package Configurations

Storing package configurations within a table on the SQL Server database computer requires a minimum amount of administrative effort. You simply specify the database in which you want to store the table and SSIS automatically creates an appropriately configured table named dbo.SSIS Configurations. Alternatively, if you have already saved one package configuration, you can save further configurations to the existing

dbo.SSIS Configurations table. It is possible to set the package configuration table to another name, but leaving it with the default name of dbo.SSIS Configurations is simpler.

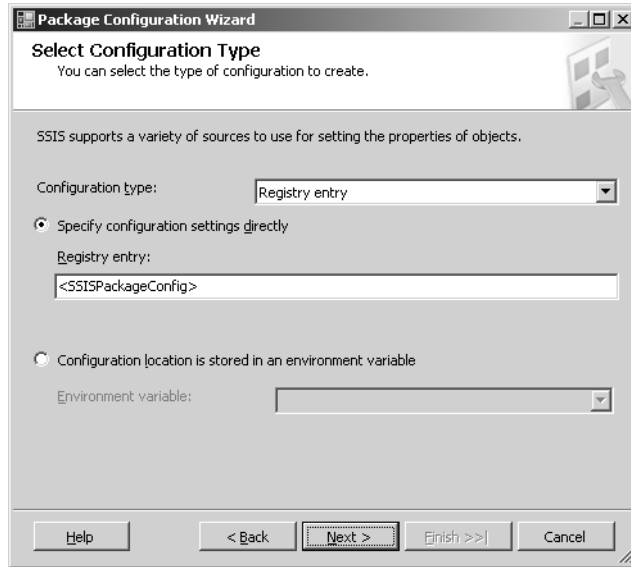


Figure 7-19 The Select Configuration Type page of the Package Configuration Wizard.

To create the dbo.SSIS Configurations table within the AdventureWorks database and save a package configuration, complete the following steps:

1. Open an existing package in BIDS.
2. Select the package in Solution Explorer and then click in the Control Flow pane.
3. From the SSIS menu, choose Package Configurations.
4. Select the Enable Package Configurations check box.
5. Click Add to start the Package Configuration Wizard and then click Next.
6. On the Select Configuration Type page, select SQL Server. Click New to specify a new connection in the Configure OLE DB Connection Manager dialog box. Click New again to create a data connection in the Connection Manager dialog box. Select a server and then below Connect To A Database, select AdventureWorks. Click OK to close the Connection Manager dialog box. Click OK to close the Configure OLE DB Connection Manager dialog box.
7. Click New next to the configuration table. Review the SQL statement that creates the table and then click OK.

8. In the Configuration Filter text box, enter a name for the Package Configuration as shown in Figure 7-20. This name will be inserted in the ConfigurationFilter column of the newly created table. Click Next.

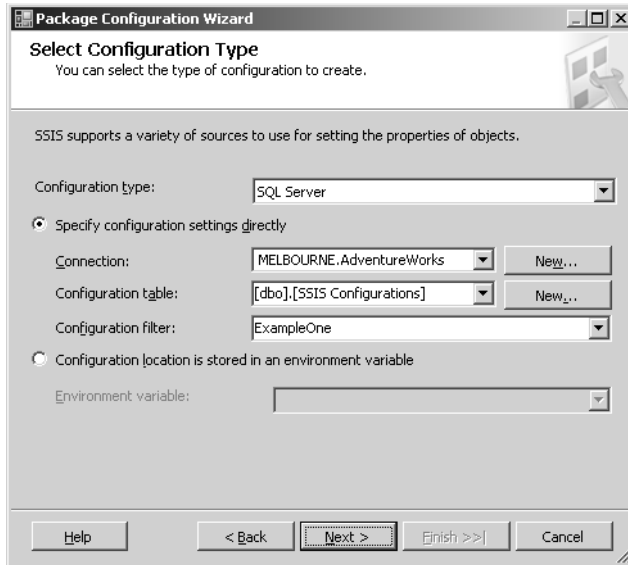


Figure 7-20 Saving package configuration to a SQL Server table.

9. Specify which properties to export to the configuration table. Select all properties in the package and then click Next.
10. Enter a configuration name. This name can be the same name that you entered in the Configuration Filter field. Click Finish, and then click Close to close the Package Configurations Organizer dialog box.

Deployment Utilities

In essence, a deployment utility is a folder that contains all files required to deploy an Integration Services project on a new server. You create a package deployment utility by configuring the build process to generate a deployment utility and then building the project. You should place support files, such as a readme.txt file, in the Miscellaneous folder of the project. When the project is built, all packages, support files, and package configurations are included and copied to the bin\Deployment folder or to a specified folder.

Creating a Package Deployment Utility

The first step in creating a deployment utility is creating a package deployment utility. To create a package deployment utility, perform the following steps:

1. Locate the project name in Solution Explorer.
2. Right-click the project and choose Properties.
3. In the Configuration Properties pane, select Deployment Utility.
4. To update package configurations when packages are deployed, set *AllowConfiguration* changes to *True*.
5. Set the *Create DeploymentUtility* property to *True*.

Building a Project

After you have created a package deployment utility, building a project is reasonably simple. Locate the project name in the Solution Explorer and right-click it. Select Build. Check the error list to see whether there were any errors. Below the error list should be a message informing you that the build succeeded.

When an Integration Services project is built, a manifest file is created and added to the bin\Deployment folder of the project. The manifest file lists all packages, package configurations, and miscellaneous files in the project. The manifest file has the name <projectname>.SSISDeploymentManifest.xml.

Deployment of Packages

To move a package to another server, you need to move the entire contents of the bin\Deployment folder to that server. You can use any method of data transfer, from shared folders to burning the contents of the folder to a DVD-ROM, carrying it to the target server, and then copying it to a temporary folder.

After you have copied the files to the target folder, you need to run the Package Installation Wizard to install them. You can invoke the Package Installation Wizard by double-clicking the <projectname>.SSISDeploymentManifest.xml file on the server on which you copied the project.

When you run the Package Installation Wizard, you are asked whether you want to deploy the SSIS package to the File System or to SQL Server, as shown in Figure 7-21. The benefits of each type of deployment were discussed in Lesson 2, “Securing SSIS Packages.” If you choose the SQL Server Deployment option, you must specify the target

server and provide the appropriate authentication. You also must choose a folder for the SSIS package dependencies to be installed in because not everything can be stored within the msdb database. The final page of the wizard provides you with a summary of the installation informing you whether the installation has occurred without error.

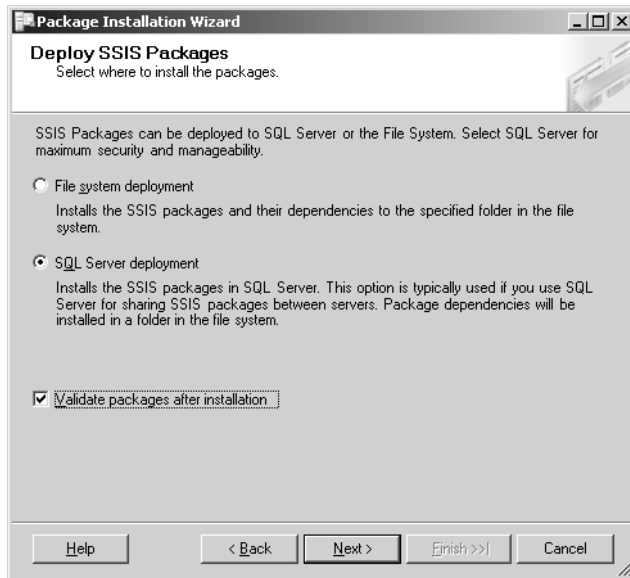


Figure 7-21 The Deploy SSIS Packages page of the Package Installation Wizard.

After you have deployed a project, modifying packages to update or extend their functionality might be necessary. If this is the case, you should rebuild the project, copy the deployment folder to the target server, and then rerun the Package Installation Wizard.

Moving a Package by Using DTUTIL

The dtutil command prompt utility can copy, move, and delete SSIS packages. You can perform these operations on packages stored within the SQL Server's msdb database, the SSIS Package Store, or the file system. The dtutil command syntax refers to these locations as /SQL /DTS and /FILE, respectively.

To copy a package stored in the msdb database on a local SQL Server 2005 computer that is configured to use Windows Authentication to the SSIS Package Store on the same computer using dtutil, issue the following command:

```
dtutil /SQL srcPackage /COPY DTS;destPackage
```

To move a package stored in the msdb database on a local SQL Server 2005 computer that is configured to use Windows Authentication to the msdb database on the local host to the msdb database on the destination host, issue the following command:

```
dtutil /SourceS localhost /SQL srcPackage /MOVE /DestS desthost SQL;destPackage
```

MORE INFO Using dtutil to delete and verify packages

For more information about how to use dtutil to delete and verify packages, as well as how to copy and move to SQL Server servers that are not configured to use Windows authentication, consult the MSDN article on the dtutil utility at [msdn2.microsoft.com/en-us/library/ms162820\(d=ide\).aspx](http://msdn2.microsoft.com/en-us/library/ms162820(d=ide).aspx).

Moving a Package by Using SQL Server Management Studio

After you have installed a package within the SSIS package store or in the msdb database on the local SQL Server computer, exporting it to another computer is relatively simple. From within SSMS, you make a connection to Integration Services. From there, all that you need to do is locate the package, right-click it, and then choose Export. This opens a dialog box that allows you to choose the export destination. The export destination can be anything from a remote instance of SQL Server 2005 to the local file system or SSIS package store. It is also possible to right-click the File System or MSDB nodes within Integration Services and choose Import. You specify the import location—which, like the export locations, can be a remote instance of SQL Server 2003, the local file system, or the SSIS package store—and then import the package.

MORE INFO Importing and exporting packages using SQL Server Management Studio

For more information about how to import and export packages by using SQL Server Management Studio, consult the MSDN article on importing and exporting packages at [msdn2.microsoft.com/en-us/library/ms141772\(SQL.90,d=ide\).aspx](http://msdn2.microsoft.com/en-us/library/ms141772(SQL.90,d=ide).aspx).

Quick Check

1. What must you do in between creating a package and deploying it?
2. Name three locations that you can copy packages to.
3. Which two places allow you to store more than one package configuration?

Quick Check Answer

1. Create a deployment utility.
2. File system, SSIS package store, MSDB database.
3. XML and database.

Scheduling Package Execution

After you have installed a package on a target server, you likely must schedule it to execute on a regular basis. You schedule the execution of SSIS packages using SQL Server Agent in SSMS. The execution of SSIS packages is one of many tasks that you can use SQL Server Agent to perform. The practice at the end of this lesson covers in more detail the process of scheduling package execution using SQL Server Agent.

You can also schedule package execution by writing a batch file that calls a package using the dtexec utility. After you've created the batch file, use the Scheduled Tasks utility in Control Panel to configure Windows to execute the batch file on a periodic basis. An advantage of this method is that you can allow Windows administrators who don't have administrative privileges within SQL Server 2005 to configure SSIS package execution schedules.

MORE INFO Using the dtexec utility

You can find out more about the dtexec utility and how it can be used to execute SSIS packages by consulting MSDN at [msdn2.microsoft.com/en-us/library/ms162810\(d=ide\).aspx](http://msdn2.microsoft.com/en-us/library/ms162810(d=ide).aspx).

PRACTICE Schedule Package Execution

In this practice, you use the SQL Server Agent service to schedule the execution of a package.

1. Open SSMS, and connect to the SQL Server on which the package is installed.
2. Locate the SQL Server Agent. (You might have to start the service if it has not been configured.)
3. To create a new job, right-click the Jobs container and choose New Job. This opens the New Job dialog box. (Alternatively, you can edit the properties of an existing job.)
4. On the General page, enter a job Name, an owner, and a category. Ensure that the Enabled check box is selected because this allows you to schedule the job.
5. Click Steps to navigate to the Steps page, and then click New. Enter a name for the Step in the Step Name field. From the Type drop-down list, select SQL Server Integration Services Package. On the General tab, open the Package Source drop-down list. Options include SQL Server, File System, or SSIS Package Store. Verify that SQL Server is selected, and then select the server on which the package is stored from the Server drop-down list. After you've selected which server the

package is on, you can click the ellipsis button (...) to select a particular package. If the package is password protected, click the Configurations tab and enter the appropriate password. Click OK to return to the Steps page.

6. Click Schedules to navigate to the Schedules page. Click New to open the New Job Schedule dialog box shown in Figure 7-22.

Figure 7-22 Job schedule properties.

7. Configure the appropriate schedule and click OK. Click OK again to create your new job.

Lesson Summary

- Package configurations are the method that SSIS uses to update the values of properties within packages at run time.
- Package configurations can be stored within an XML configuration file, an environment variable, a registry key, a SQL Server table, or the parent package's configuration.
- A deployment utility is a folder that contains all files required to deploy an Integration Services project on a new server.
- To move a package to another server, move the entire contents of a project's bin\Deployment folder to that server.
- The execution of SSIS packages is scheduled using SQL Server Agent.

Lesson Review

You can use the following questions to test your knowledge of the information in Lesson 4, “Deploying SSIS Packages.” The questions are also available on the companion CD if you prefer to review them in electronic form.

NOTE Answers

Answers to these questions and explanations of why each answer choice is correct or incorrect are located in the “Answers” section at the end of the book.

1. Which of the following package configuration file locations can store more than one package configuration file? (Choose all that apply.)
 - A. XML configuration file
 - B. Environment variable
 - C. Registry entry
 - D. SQL Server table
2. When using Solution Explorer, in which folder should you place support files such as readme.txt?
 - A. Data Sources
 - B. Data Source Views
 - C. SSIS Packages
 - D. Miscellaneous
3. Which of the following utilities would you use to schedule the execution of a package? (Choose all that apply.)
 - A. Business Intelligence Development Studio
 - B. SSIS Package Installation Wizard
 - C. SQL Server Agent
 - D. Scheduled Tasks
4. Which of the following utilities is directly used to install an SSIS package on a production SQL Server 2005 computer?
 - A. Add/Remove programs in Control Panel
 - B. SQL Server Management Studio
 - C. Package Installation Wizard
 - D. Business Intelligence Development Studio

Chapter Review

To further practice and reinforce the skills you learned in this chapter, you can complete the following tasks:

- Review the chapter summary.
- Review the list of key terms introduced in this chapter.
- Complete the case scenarios. These scenarios set up real-world situations involving the topics of this chapter and ask you to create a solution.
- Complete the suggested practices.
- Take a practice test.

Chapter Summary

- A package is an organized collection of tasks and workflow elements that allow for the automation of complex processes within SQL Server 2005.
- Creating SSIS packages involves collecting control flow and data flow tasks from a set of toolboxes within the Business Intelligence Design Studio IDE.
- It is possible to store packages, package configurations, and log files within specific SQL Server databases using appropriate database and table level security. Any packages stored within the file system should be protected with appropriate file system security.
- The primary method for restarting failed packages is to institute checkpoints, which allow task host containers, Foreach Loop containers, and transacted containers to be rerun from the point of failure. Transactions can be used to allow a group of tasks in a package to be linked as a single unit.
- Packages can be debugged visually using the debugging menu in the BIDS IDE. Data viewers can be added, allowing the developer to add viewers that display data transformations in real time.
- Logging can be configured at the package, task, or container level. Logs can be written to SQL Server Profiler, SQL Server, the Windows Event Log, or XML files.
- Package configurations are the method that SSIS uses to update the values of properties within packages at run time, and these configurations are stored within an XML configuration file, an environment variable, a registry key, a SQL Server table, or the parent package's configuration.

Key Terms

Do you know what these key terms mean? You can check your answers by looking up the terms in the glossary at the end of the book.

- checkpoints
- configurations
- control flow elements
- data flow elements
- logging
- Package Deployment utility
- SSIS variables
- transaction

Case Scenarios

In the following case scenarios, you will apply what you've learned about designing and managing SSIS packages. You can find answers to these questions in the "Answers" section at the end of this book.

Case Scenario 1: Creating and Managing SSIS Packages

You are a systems administrator for Contoso's mineral exploration division. You are currently preparing Contoso's remote exploration vehicle, a modified all-terrain truck that will be sent deep into the Western Australian desert for several months looking for uranium deposits. The truck has a pair of computers running SQL Server 2005 and a third computer that works as a file server. Because the truck will often be hundreds of miles from any settlement, it will communicate with Contoso's corporate base using a low-bandwidth satellite uplink. You are currently developing SSIS packages so that many of the vehicle's SQL Server data collection and reporting tasks can be automated.

1. You need to create a task that will copy files from each SQL Server server to the file server, send an e-mail to you if either the copy or the transfer task fails, transfer files over the satellite uplink's Internet connection, and send an e-mail to you if some aspect of the transfer via satellite fails. You review the various tools available in the Control Flow Items toolbox within Business Intelligence Design Studio. Which tools would you use in constructing this package?

2. Each time the SSIS package responsible for transferring files over the exploration vehicle's satellite link to the Internet executes, 25 files are uploaded. Occasionally the transfer of one or two of these files fails. You want to ensure that only files that don't successfully transfer are uploaded if the package is reexecuted. How could you ensure this occurs?
3. The exploration truck's staff will have Windows Administrator privileges, but it will have no direct access to SQL Server 2005 itself. Which event log provider should you configure SSIS packages to write events to so that the exploration truck staff will be informed if package execution fails?

Case Scenario 2: SSIS Package Administration

You are working with the SQL Server development team at Tailspin Toys. The development team's computers are all located in the `development.tailspintoys.internal` domain, a part of the Tailspin Toys Windows Server 2003 functional level forest. The SQL Server development team has two computers running SQL Server 2005 Developer edition that are used only for developmental purposes. Two computers in the domain `production.tailspintoys.internal` are running SQL Server 2005 Enterprise Edition. The developers don't have any direct access to these computers, and you are responsible for the installation of any SSIS packages the development team produces. The environment has the following properties:

- All SQL Server servers in the Tailspin Toys forest are configured to use Windows authentication.
- All packages are currently stored within the `msdb` databases on the computers running SQL Server 2005 Developer Edition.
- Several of the administrators within the production domain do not have access to SQL Server 2005 but will be responsible for scheduling package execution.

Answer the following questions:

1. Describe how the administrators in the production domain without SQL Server 2005 privileges could schedule package execution.
2. Which technology can you use to ensure that packages aren't modified after the developers have completed their work on them?
3. Describe three different methods that you could use to copy packages from the development SQL Server 2005 Developer Edition computer to the two SQL Server 2005 Enterprise Edition computers in the production domain.

Suggested Practices

To help you successfully master the exam objectives presented in this chapter, complete the following practice tasks.

Design and Manage SQL Server Integration Services (SSIS) Packages

- **Practice 1: Create a basic package.** Use BIDS to create a package that executes Solitaire, Notepad, and then Calculator.
- **Practice 2: Create an advanced package.** Create a package that determines how much money AdventureWorks has made from each product they sell, sort it in descending order, and write the result to a text file.
- **Practice 3: Secure an advanced package.** Secure the package that you created in Practice 2 with a password.
- **Practice 4: Configure restart points on an advanced package.** Configure restart points on the package that you created in Practice 2.
- **Practice 5: Create a package configuration.** Create a package configuration for the package that you created in Practice 2, and store this configuration within the AdventureWorks database.

Take a Practice Test

The practice tests on this book's companion CD offer many options. For example, you can test yourself on just one exam objective, or you can test yourself on all the 70-444 certification exam content. You can set up the test so that it closely simulates the experience of taking a certification exam, or you can set it up in study mode so that you can look at the correct answers and explanations after you answer each question.

MORE INFO Practice tests

For details about all the practice test options available, see the "How to Use the Practice Tests" section in this book's Introduction.

Chapter 8

Design Data Integrity

At its simplest, designing data integrity is about guaranteeing the quality of the data that resides within the database. The quality and utility of a database is directly proportional to the quality of the data it hosts in its tables. If some of the data stored in the tables is rubbish, confidence in the accuracy of the information provided by the database decreases.

Bad data enters the database in several ways. The most obvious method is that one of the people doing the grunt work of entering data makes an error that isn't discovered. This error remains in the database until someone else notices it and manually corrects it. Another form of error occurs when updates to database data conflict and the database has to arbitrarily decide that one update will have precedence over another. Learning how to use the internal process of Microsoft SQL Server 2005 to resolve these sorts of conflicts and how to ensure that data is formatted correctly and has the appropriate properties forms the basis of the lessons in this chapter.

Exam objectives in this chapter:

- Design data integrity.
 - Reconcile data conflicts.
 - Make implicit constraints explicit.
 - Assign data types to control characteristics of data stored in a column.

Lessons in this chapter:

- Lesson 1: Reconciling Data Conflicts. 421
- Lesson 2: Making Implicit Constraints Explicit. 428
- Lesson 3: Assigning Data Types to Control Characteristics of Data Stored in a Column. 442

Before You Begin

To complete the lessons in this chapter, you must have completed the following tasks:

- Configured a Microsoft Windows Server 2003 R2 computer with SQL Server 2005 Enterprise Edition SP1 as detailed in the Appendix.
- Installed an updated copy of the AdventureWorks sample database as detailed in the Appendix.

No additional configuration is required for this chapter.

Lesson 1: Reconciling Data Conflicts

Data conflicts are inevitable when it is possible to modify data in replicated databases at multiple locations. This inevitability stems from the latency inherent in replication between a publisher and its subscribers. The possibility always exists that during that replication window two separate people at two separate locations might alter the same piece of information. This lesson discusses the processes that you can configure within SQL Server to resolve this type of conflict.

NOTE Replication

Although Chapter 10, “Replication,” more explicitly details the intricacies of the SQL Server 2005 replication process, understanding this lesson requires only that you grasp the concepts of Publisher, Subscriber, and Article, which are key components of this process. A *Publisher* is a database instance that makes data available to other locations through replication. A *Subscriber* is a database instance that receives replicated data from one or more Publishers. Subscribers are able to forward changes back to Publishers or republish data to other Subscribers. An *Article* is a database object, such as a table, included in a publication.

After this lesson, you will be able to:

- Understand how SQL Server 2005 detects data conflicts.
- Understand the different conflict resolution policies available in SQL Server 2005.
- Configure a conflict resolution policy.

Estimated lesson time: 30 minutes

Detecting Conflicts

When you create a publication, the replication process adds a Uniqueidentifier column to all tables included in the Article, as shown in Figure 8-1. Whenever published data is modified at either the Publisher or the Subscriber, SQL Server inserts a new globally unique identifier (GUID) in the Uniqueidentifier rows where the modified data resides. When synchronization between the Publisher and Subscriber occurs, the Queue Reader Agent compares the previous and current values of these GUIDs to determine whether a conflict exists.

During synchronization, a transaction in the queue maintains both the old and new rows of GUIDs. The two GUIDs in the transaction and the GUID in the relevant table on the publication are compared when the transaction is applied at the Publisher. If the old GUID stored within the transaction is identical to the GUID in the publication, SQL Server updates the publication. When this update occurs, the appropriate table

row's Uniqueidentifier column is assigned the GUID that was generated by the Subscriber when the change was made. The old GUID matching the publication's GUID indicates that the publication's data for that row has not changed since the data at the Subscriber was last synchronized.

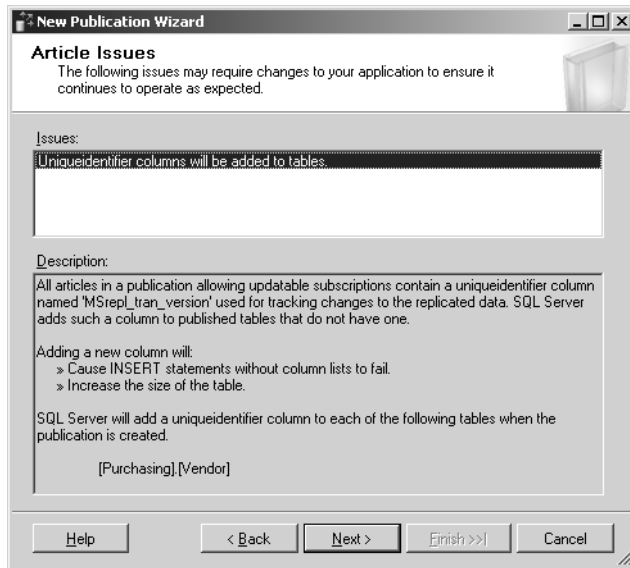


Figure 8-1 The addition of Uniqueidentifier columns to tables.

If the old GUID stored in the transaction does not match the publication's GUID, SQL Server detects a conflict. A mismatch indicates that two different versions of the row exist, one in the transaction submitted by the Subscriber and a newer one that exists on the Publisher. This happens when another Subscriber updates the same row prior to the existing Subscriber's transaction being synchronized. For example, Server A is configured as a Publisher for Server B and Server C. Updates are made to the same data on both Server B and Server C at the same time. Server B and Server C synchronize with Server A. Server C gets in first, its "old Server A row data GUID" matches the existing "Server A row data GUID," and its newly generated GUID gets assigned on Server A. When Server B synchronizes, its "older Server A row data GUID" doesn't match the newly updated Server A GUID and a conflict is detected. What occurs next is dependent on the conflict resolution policy that you configured as a part of the publication.

Resolving Conflicts

You select a conflict resolution policy when you create a publication that uses queued updating. The conflict resolution policy determines how the Queue Reader Agent treats different versions of the same row during the synchronization process. There

are three possible conflict resolution policies:

- Keep The Publisher Change
- Reinitialize The Subscription
- Keep The Subscriber Change

Keep The Publisher Change is the default policy. You can alter the conflict resolution policy only if there are no existing subscriptions. You alter the conflict resolution policy by performing the following steps:

1. Open SQL Server Management Studio, and connect to the database instance that is functioning as a Publisher.
2. Expand the Replication node.
3. Expand the Local Publications node.
4. Right-click the publication for which you want to alter the conflict policy and choose Properties. This opens the Publication Properties dialog box.
5. In the Select A Page pane, select Subscription Options.
6. Use the drop-down list next to Conflict Resolution Policy, as shown in Figure 8-2, to select a policy.

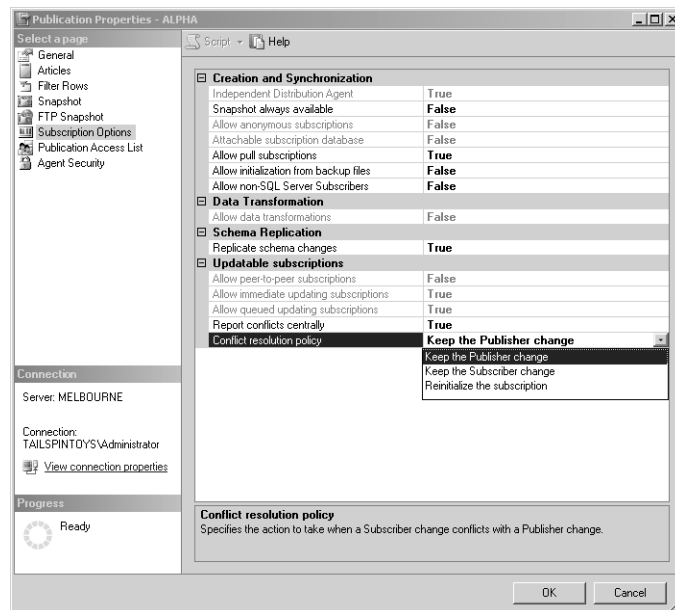


Figure 8-2 Selecting a conflict resolution policy.

The remainder of this section explains how each of the three conflict resolution policies work.

Keep The Publisher Change

If you apply the Keep The Publisher Change conflict resolution policy to a publication, SQL Server maintains consistency based on the data that exists at the Publisher. If a conflict arises, the conflicting transaction rolls back at the Subscriber. SQL Server then updates the rows on the Subscriber that caused the conflict to match the rows on the Publisher. For example, Server A is configured as a Publisher. Server B and Server C are configured as Subscribers. A change is made to the same row of data on Server B and Server C at the same time by different people. Server C synchronizes with Server A first. When synchronization is complete, Server A holds Server C's updated data. Server B then attempts to synchronize with Server A, but the Queue Reader Agent detects a conflict. Detecting the conflict sets off a process that rolls back the data change on Server B and then rewrites the row data on Server B with the new data on Server A. The conflict resolution process in this case has given precedence to the update that occurred on Server C rather than the update that occurred on Server A. If Server B had synchronized with the Publisher before Server C had, the update on Server B would have had precedence and would have replicated to all servers.

Reinitialize The Subscription

When you implement the Reinitialize The Subscription policy on a publication, the detection of a conflict causes the rejection of all transactions in the queue, including the transaction that caused the conflict. This option has wider implications than the default policy, as other nonconflicting updates are lost as well. For example, an update on Subscriber Server B alters data in the first two rows of a table. An update at the same time on Subscriber Server C alters data in the first row of the same table. Server C synchronizes with the Publisher first, and the first row of the table on the Publisher is updated. When Server B attempts to synchronize, the Queue Reader Agent detects a conflict and rejects all transactions. Both the first and the second row alterations made on Server B are lost.

Keep The Subscriber Change

When you apply the Keep The Subscriber Change policy to a publication, the last Subscriber transaction to update the Publisher is retained. When a conflict is detected by the Queue Reader Agent, the transaction sent by the Subscriber is used and the Publisher is updated. In the example just discussed, where Server B and Server C have

the same data updated and Server C synchronizes first, the update from Server B replaces the update from Server C on the Publisher because it is the last to be applied.

MORE INFO Understanding conflict resolution

For more information about queued updating conflict detection and resolution, consult the following MSDN article: [msdn2.microsoft.com/en-us/library/ms151177\(SQL.90,d=ide\).aspx](https://msdn2.microsoft.com/en-us/library/ms151177(SQL.90,d=ide).aspx).

Quick Check

1. What are the names of the three conflict resolution policies?
2. What is the name of the component that detects conflicts?
3. Which conflict resolution policy is in force if SQL Server dumps the transaction queue when it detects a conflict?

Quick Check Answer

1. The names of the three conflict resolution policies are Keep The Publisher Change, Keep The Subscriber Change, and Reinitialize The Subscription.
2. The Queue Reader Agent is the component that detects conflicts.
3. The Reinitialize The Subscription policy is used in this situation.

Exam Tip Use your scratch paper. Almost all exam providers give you some working paper or a quick-erase board to help you with your exam. Many test takers are too stressed by the experience to take proper advantage of this resource. Sometimes, question authors provide you with so much information that you can easily get your facts confused. For example, you might read about columns in several different tables that have similar names. In the exam environment, it can be easy to confuse the columns and their descriptions when presented with more than one or two. Take a moment and jot down the column names and descriptions on your working paper. Check them against the question text to make sure that you got them right and then start working on the question. Working off your own list is easier than going back into a paragraph of question text to work out whether you've identified the correct items.

Viewing Data Conflicts

You can view conflict data using the Microsoft Replication Conflict Viewer. Conflict data is stored until the end of the conflict retention period, which defaults to 14 days. You can alter the conflict retention period by specifying a retention value for the *@conflict_retention* parameter of the *sp_addpublication* stored procedure.

If there are no conflicts, the Replication Conflict Viewer reports in a dialog box that no conflicts exist and you are unable to proceed further. If conflicts do exist, the Replication Conflict Viewer allows you to filter rows, select and remove rows from the conflict's metadata table, and export the details of a conflict by selecting Log The Details Of This Conflict and entering a file name for the file where details will be written.

MORE INFO Viewing data conflicts

For more information about viewing data conflicts using Microsoft Replication Conflict Viewer, consult the following MSDN article: [msdn2.microsoft.com/en-us/library/ms151865\(SQL.90,d=ide\).aspx](http://msdn2.microsoft.com/en-us/library/ms151865(SQL.90,d=ide).aspx).

PRACTICE Viewing Conflicts

When conflicts occur, you should look over them to see that important data has not been lost during the conflict resolution process. Regularly reviewing conflict data helps you to determine whether you are enforcing the appropriate conflict resolution policy. To use Microsoft Replication Conflict Viewer to review conflict data, perform the following steps:

1. Open SSMS, and connect to the Publisher's instance.
2. Expand the Replication folder.
3. Expand the Local Publications folder.
4. Right-click the publication you want to examine for conflicts and select View Conflicts. A dialog box informs you if no conflicts are present.
5. Click OK.

Lesson Summary

- Conflicts occur when updates to the same data occur at different Subscribers at the same time and those subscribers then synchronize with the Publisher.
- The Queue Reader Agent is the SQL Server subsystem that detects conflicts.
- Conflict resolution policies determine how the Publisher deals with conflicts.
- If the Keep The Publisher Change conflict resolution policy is enforced, when a conflict occurs, the transaction is rolled back at the Subscriber and the Subscriber data is overwritten by the data on the Publisher.
- If the Reinitialize The Subscription conflict resolution policy is enforced, the detection of a conflict will cause the rejection of all transactions in the queue, including the transaction that caused the conflict.

- If the Keep The Subscriber Change conflict resolution policy is enforced, the last Subscriber transaction to update the Publisher is retained.
- You can view conflict data using the Microsoft Replication Conflict Viewer. Conflict data is stored until the end of the conflict retention period (a default of 14 days).

Lesson Review

You can use the following questions to test your knowledge of the information in Lesson 1, “Reconciling Data Conflicts.” The questions are also available on the companion CD if you prefer to review them in electronic form.

NOTE Answers

Answers to these questions and explanations of why each answer choice is correct or incorrect are located in the “Answers” section at the end of this book.

1. Which conflict resolution policy ensures that the publisher is always updated when a Subscriber synchronizes with new data?
 - A. Keep The Publisher Change
 - B. Reinitialize The Subscription
 - C. Keep The Subscriber Change
2. Which SQL Server 2005 subsystem detects data conflicts during Subscriber/Publisher synchronization?
 - A. SQL Server Integration Services
 - B. SQL Server Agent
 - C. Queue Reader Agent
 - D. SQL Server Analysis Services
3. Ian enters some changes to correct a row on a Subscriber database. When he checks back an hour later, he notices that the row has changed but that the changes are different from the ones that he entered. He checks and finds that a conflict occurred. Which of the following conflict resolution policies might apply to the publication? (Choose all that apply.)
 - A. Keep The Publisher Change
 - B. Reinitialize The Subscription
 - C. Keep The Subscriber Change

Lesson 2: Making Implicit Constraints Explicit

As you might remember from the study you have undertaken for previous SQL Server 2005 exams, *constraints* are limitations on the data that users can insert in a column of a database table. UNIQUE or PRIMARY KEY constraints prevent users from inserting a value that already exists within the column, and CHECK constraints prevent users from inserting values that do not meet a particular rule. In the context of SQL Server 2005, an *implicit constraint* is a business rule. Implicit constraints could be anything from “no contractor can bill more than 60 hours in a week” to the maximum number of items that a customer can order at any one time. As a database administrator, you might be responsible for ensuring that the database reflects the policy rules of the business. An *explicit constraint* is one that the administrator applies to the database. Following from the example just mentioned, the database administrator would implement an explicit constraint by programming the database so that it would be impossible for a contractor to enter a value higher than 60 hours per week into the organization’s billing database.

After this lesson, you will be able to:

- Make implicit constraints explicit.
- Define primary key constraints.
- Define unique constraints.
- Define check constraints.
- Define foreign key constraints.
- Apply default definitions to columns.
- Configure the nullability of columns.
- Define triggers.

Estimated lesson time: 30 minutes

Understanding Implicit and Explicit Constraints

You can think of an implicit constraint as a business or organizational rule. An example might be that all employees within a particular department always have a salary range between 40 and 60 thousand dollars a year. To make this implicit constraint explicit, you must devise a way to ensure that the database does not allow users to enter values outside this range.

Constraints

Constraints determine whether users can store particular data in a column depending on a particular rule. This rule might be as simple as “allow any value to be stored in this column except values that are already stored within this column.” Constraints can also be more complex, allowing users to insert into a table’s column only values that conform to a particular logical rule. The following types of constraints are covered over the next few pages:

- PRIMARY KEY
- UNIQUE
- FOREIGN KEY
- CHECK

PRIMARY KEY Constraints

A PRIMARY KEY constraint enforces the uniqueness of data, and it stops the insertion of null values into columns that are included within the constraint. A table can have only one PRIMARY KEY constraint. A column that has a PRIMARY KEY constraint applied cannot contain null values. When you specify a PRIMARY KEY constraint for a table, the database engine enforces data uniqueness by creating a unique index for the primary key column.

You can apply PRIMARY KEY constraints to multiple columns. When you do this, the combination of values from all columns within the constraint definition must be unique. So it is possible to apply a PRIMARY KEY constraint to the FirstName and LastName columns even if the last name column has duplicate entries. It would not be possible to do this with these two columns if two people had the same first name and last name, though it might be possible to add a third column containing the middle initial and so on.

To create a PRIMARY KEY constraint on an existing table, perform the following steps:

1. Open SSMS, and connect to the database instance that hosts the table on which you want to insert the PRIMARY KEY.
2. Expand the Databases folder to locate the database that hosts the table on which you want to insert the PRIMARY KEY.

3. Expand the appropriate database, and locate the table on which you want to insert the PRIMARY KEY.
4. Right-click the table and choose Modify.
5. Ensure that the Allow Nulls check box on the column on which you want to apply the PRIMARY KEY is not selected.
6. Right-click the column and choose Set Primary Key from the menu that appears, which is shown in Figure 8-3.

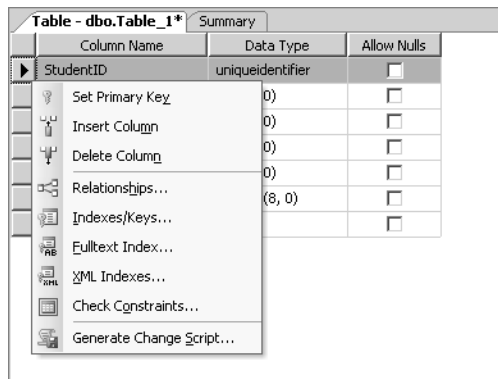


Figure 8-3 Configuring a Primary Key.

7. On the toolbar, click Save to save your changes to the table.

UNIQUE Constraints

You can apply a UNIQUE constraint to a column to ensure that entering duplicate values is impossible. UNIQUE constraints, like PRIMARY KEY constraints, can be applied to multiple columns, such as in the FirstName and LastName example discussed earlier. The primary difference between the two is that you can configure multiple UNIQUE constraints to a table, whereas you can configure only a single PRIMARY KEY constraint for a table. UNIQUE constraints also allow users to enter a single NULL value. UNIQUE constraints can be referenced by FOREIGN KEY constraints. You cannot apply a UNIQUE constraint to a column when existing duplicate values are in the target column.

To create a UNIQUE constraint on a column in an existing table, perform the following steps:

1. Open SSMS, and connect to the database instance that hosts the table that has the column to which you want to apply the UNIQUE constraint.

2. Expand the Databases folder to locate the database that hosts the table that has the column to which you want to apply the UNIQUE constraint.
3. Expand the appropriate database, and locate the table that has the column to which you want to apply the UNIQUE constraint.
4. Right-click the table and choose Modify.
5. Right-click the column to which you want to apply the UNIQUE constraint and choose Indexes/Keys.
6. In the Indexes/Keys dialog box, click Add. By default, this creates a new Index, but you will change this to a UNIQUE constraint.
7. In the General area in the Columns field, select the column to which you want to apply the UNIQUE constraint.
8. Change the Is Unique field to Yes.
9. Change the Type to Unique Key.
10. In the Identity box, change the name to something like Unique_ColumnName. An example is shown in Figure 8-4.
11. Click Close.

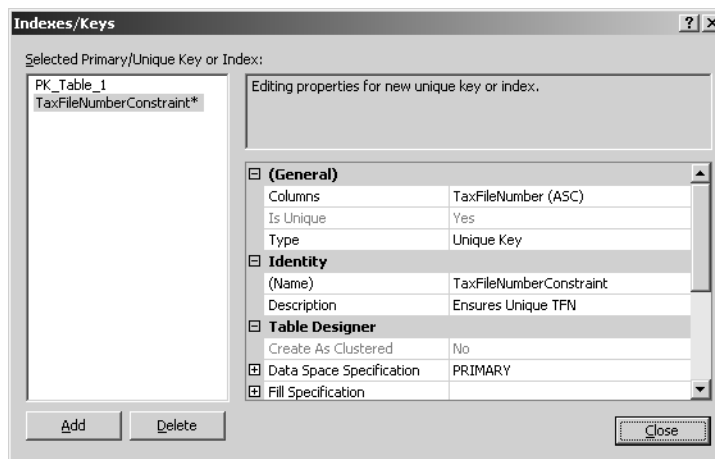


Figure 8-4 Configuring a Unique constraint.

FOREIGN KEY Constraints

In general, you use a FOREIGN KEY constraint to link data in two separate tables, though it is possible for a FOREIGN KEY constraint to reference another column in the same table. FOREIGN KEY constraints can apply to more than one column and

must be linked to columns that either have a PRIMARY KEY or UNIQUE constraint applied.

You can use FOREIGN KEY constraints to enforce referential integrity. When referential integrity is enforced, data in the foreign table is protected from deletion if that deletion orphans data in the referencing table. For example, in the AdventureWorks database there are two tables, Sales.SalesPerson and Sales.SalesOrderHeader, that are linked using a FOREIGN KEY constraint. The SalesPersonID column of the Sales.SalesOrderHeader table is linked using a FOREIGN KEY constraint to the SalesPersonID column of the Sales.SalesPerson table. Referential integrity ensures that a salesperson's row cannot be deleted from the Sales.SalesPerson table as long as any row in the Sales.SalesOrderHeader table makes reference to that salesperson's ID. When SQL Server enforces referential integrity, you can delete a row only from a table that is the target of a FOREIGN KEY constraint if the data in that row is not referenced in the table to which you have applied the FOREIGN KEY constraint.

NOTE Limit on number of FOREIGN KEY constraints

Although SQL Server does not have any default limit on the number of FOREIGN KEY constraints that you can apply to a table, the number of constraints is limited by hardware configuration and database design. Microsoft recommends that no table contain more than 253 FOREIGN KEY constraints and that no single table should be referenced by more than 253 FOREIGN KEY constraints.

To create a FOREIGN KEY constraint, perform the following steps:

1. Open SSMS, and connect to the database instance that hosts the table that has the column to which you want to apply the FOREIGN KEY constraint.
2. Expand the Databases folder to locate the database that hosts the table that has the column to which you want to apply the FOREIGN KEY constraint.
3. Expand the appropriate database, and locate the table that has the column to which you want to apply the FOREIGN KEY constraint.
4. Right-click the table and choose Modify.
5. From the Table Designer menu, choose Relationships.
6. In the Foreign-Key Relationships dialog box, click Add.
7. Click Tables And Columns Specification in the grid to the right, and then click the ellipses (...) to the right of the property.
8. In the Tables And Columns dialog box, in the Primary Key Table drop-down list, select the table that will be on the primary or unique key side of the relationship.

9. In the grid beneath the Primary Key Table text box, select the column that will be used as the table's primary key.
10. In the adjacent grid cell, select the column in the table that will be using the foreign key, as shown in Figure 8-5.

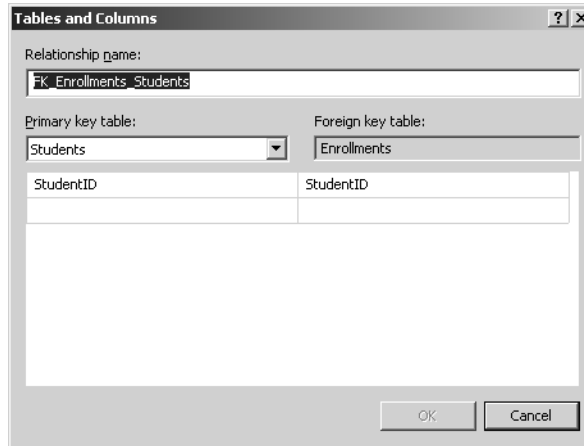


Figure 8-5 Configuring a FOREIGN KEY constraint.

11. Click OK to create the relationship, and then click Close to close the Foreign Key Relationships dialog box.

Real World

Orin Thomas

When I first got my driver's license, the person entering the data to be printed on the license kept misspelling the name of the suburb in which I lived. After the third error, I decided to put up with the mistake and hope that when I renewed my license ten years later the person responsible for entering the data would know how to spell. This would not have been a problem if the license database had some sort of FOREIGN KEY constraint linking the suburb I lived in to a list of all localities within my state as a way of enforcing data integrity. If a FOREIGN KEY had been in place, only an approved spelling of my suburb's name would have been able to have been used on my license. Alas, the technology wasn't sophisticated enough in those days, and I often had to explain that some of the information on my primary form of identification was spelled incorrectly.

CHECK Constraints

CHECK constraints enforce data integrity by limiting the values that users can enter into a column. CHECK constraints determine whether data is valid by applying a logical expression. You can create CHECK constraints with any logical expression that returns TRUE or FALSE based on the logical operators. For example, if you want to ensure that values in the temperature column are limited to values between -5 and 150, you should use the following logical expression:

```
temperature >= -5 AND temperature <= 150
```

It is possible to apply multiple CHECK constraints to a single column. It is also possible to apply a single CHECK constraint to several columns within a table. To do this, you must create the CHECK constraint at the table level.

CHECK constraints have a problematic relationship with null values because null values evaluate to UNKNOWN. If you have a column that is of type int to which you have applied a CHECK constraint that requires the column value to equal 20, and you then insert the value NULL into the column, the CHECK constraint does not return an error because it can't evaluate the value.

To create a CHECK constraint on a table, perform the following steps:

1. Open SSMS, and connect to the database instance that hosts the table that has the column to which you want to apply the CHECK constraint.
2. Expand the Databases folder to locate the database that hosts the table with the column to which you want to apply the CHECK constraint.
3. Expand the appropriate database, and locate the table with the column to which you want to apply the CHECK constraint.
4. Right-click the table and choose Modify.
5. Right-click the column to which you want to apply the CHECK constraint and choose Check Constraints. This opens the Check Constraints dialog box as shown in Figure 8-6.
6. Click Add to add a constraint.
7. In the grid, in the Expression field, type the SQL expression of the CHECK constraint. See the More Info box for more information about creating SQL expressions for CHECK constraints.
8. Click Close.

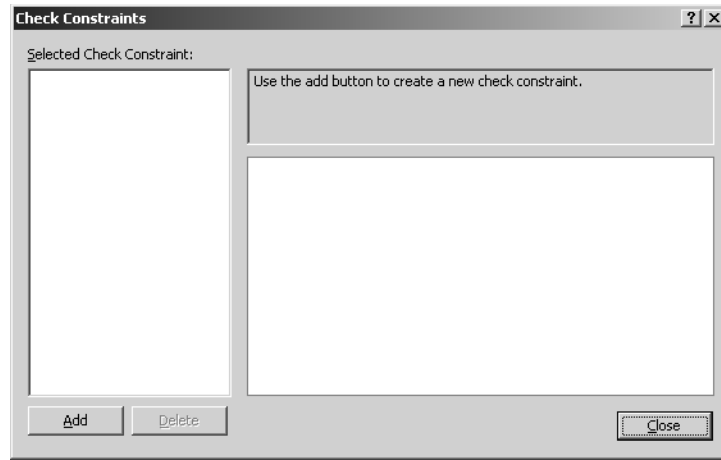


Figure 8-6 Adding a Check constraint.

MORE INFO Writing CHECK constraints

For more information about how to write a CHECK constraint, consult the following MSDN article: [msdn2.microsoft.com/en-us/library/ms191245\(SQL.90,d=ide\).aspx](https://msdn2.microsoft.com/en-us/library/ms191245(SQL.90,d=ide).aspx).

DEFAULT Definitions

In SQL Server 2005, each column in a record must contain a value, even if that value is NULL. If a column does not allow for null values, you must explicitly specify a value for the column or the database engine returns an error. Because the value NULL might not be desirable and you might need to load a row of data into a table when you do not yet know some column values, you can apply a DEFAULT definition to the column. When you insert a row into a table but don't provide data for the column to which the DEFAULT definition applies, SQL Server automatically inserts the value in the DEFAULT definition.

NOTE Common default definitions

It is common practice to specify zero as the DEFAULT definition for numeric columns and N/A as the DEFAULT definition for string columns.

To create a DEFAULT definition for a column in a table, perform the following steps:

1. Open SSMS, and connect to the database instance that hosts the table that has the column to which you want to apply the DEFAULT definition.

2. Expand the Databases folder to locate the database that hosts the table that has the column to which you want to apply the DEFAULT definition.
3. Expand the appropriate database, and locate the table with the column to which you want to apply the DEFAULT definition.
4. Right-click the table and choose Modify.
5. Select the column for which you want to specify a default value.
6. In the Column Properties tab, enter the new default value in the Default Value Or Binding property as shown in Figure 8-7.

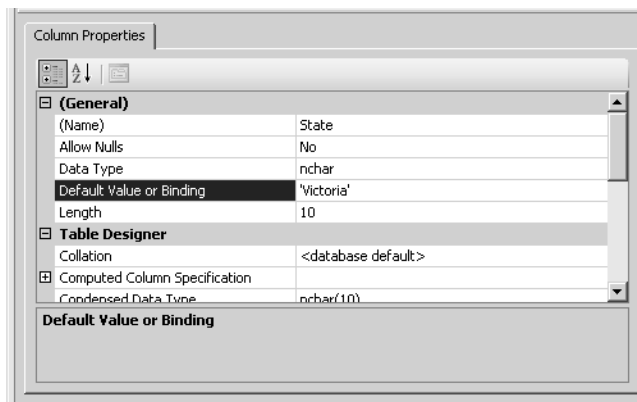


Figure 8-7 Setting the default value of the State column to Victoria.

Allowing NULL Values

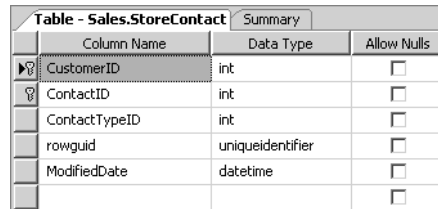
In certain circumstances, you might want to configure a column to allow NULL values. It is important to realize that a value of NULL is different from zero or a zero-length character such as "". A value of NULL signifies that no entry has been made into a column. SQL Server inserts a value of NULL if a user inserts a row without including a value for a specific column, NULL values are allowed, and no DEFAULT definition exists.

Allowing NULL values is limiting, because if a column has NULL values, you cannot apply a PRIMARY KEY constraint to it. If there is more than one NULL value in a column, you cannot apply a UNIQUE key constraint to it. It also follows that if there is more than one NULL value in a column, it cannot be the target of a FOREIGN KEY constraint.

By default, columns permit NULL values. To modify the NULL option of a column, perform the following steps:

1. Open SSMS, and navigate to the database that contains the table with the column you want to modify.

2. Right-click the appropriate table in Object Explorer and choose Modify.
3. Select the appropriate column, and clear the Allow Nulls check box as shown in Figure 8-8.



Column Name	Data Type	Allow Nulls
CustomerID	int	<input type="checkbox"/>
ContactID	int	<input type="checkbox"/>
ContactTypeID	int	<input type="checkbox"/>
rowguid	uniqueidentifier	<input type="checkbox"/>
ModifiedDate	datetime	<input type="checkbox"/>

Figure 8-8 Configuring nullability.

You can change the properties of an existing column only if it contains no existing NULL values and there is no index on the column. To disallow NULL values in an existing column that contains NULL values, you should perform the following steps:

1. Create a new column that uses a DEFAULT definition to insert an appropriate value.
2. Copy the data from the existing column to the new column.
3. Delete the old column.
4. Rename the new column to the old column's name.

Quick Check

1. Which type of constraint would you use to ensure that users can insert only numerical values that are within a particular range into a table?
2. How can you ensure that a value of 0 is applied to a column of type int if an INSERT statement does not specify a value for the column?
3. What is the maximum number of NULL values that can exist in a column if you want to apply a UNIQUE constraint to it?

Quick Check Answer

1. You would use the CHECK constraint to accomplish this.
2. Use the DEFAULT definition.
3. One is the maximum number allowed in this situation.

Triggers

DML triggers operate on UPDATE, INSERT, and DELETE statements. You can configure them to enforce business rules and enforce data integrity when users modify data in tables or views. You create, modify, and drop DML triggers using Transact-SQL syntax. DML triggers can perform the following functions:

- They can cascade changes through related tables in the database.
- They can guard against incorrect INSERT, UPDATE, and DELETE operations and enforce restrictions that are more complicated than those that you can define using CHECK constraints.
- They can be used to evaluate a table's state before and after data modification and use that evaluation to initiate actions.
- Multiple triggers of the same type (INSERT, UPDATE, or DELETE) applied to a table allow different actions to be performed depending on the content of the statement that fires the trigger.

There are three types of DML triggers: AFTER, INSTEAD OF, and CLR.

AFTER Triggers

SQL Server executes AFTER triggers after it performs the action in an INSERT, UPDATE, or DELETE statement. You can specify AFTER triggers only on tables.

INSTEAD OF Triggers

SQL Server executes INSTEAD OF triggers in place of the usual triggering action. Rather than SQL Server inserting a new row in the table, as was the case with the previous example, the INSTEAD OF trigger fires before SQL Server makes any modifications based on the triggering statement. For example, an INSTEAD OF trigger could fire if a DELETE statement is detected. Rather than deleting data from the table, the INSTEAD OF trigger performs another action. INSTEAD OF triggers are also often used to extend the types of updates that a view can support. For example, INSTEAD OF triggers can be configured to modify multiple base tables that contain columns using the timestamp data type.

CLR Triggers

A common language runtime (CLR) trigger can be either an AFTER or INSTEAD OF trigger. CLR triggers execute methods written in managed code. You use these triggers when you require more complex actions than you can achieve by using Transact-SQL.

MORE INFO Creating triggers

Creating a trigger is a detailed process. You can learn more about creating triggers by reviewing the following MSDN article: msdn.microsoft.com/library/default.asp?url=/library/en-us/tsqlref/ts_create2_7eeq.asp.

Exam Tip Although it might not always seem that way, certification exam questions are not written as tricky riddles. The goal of each exam question is to test your understanding of a concept. If you completely understand the concepts behind a question, the answer becomes obvious. Sometimes you can get a better handle on a question by asking yourself, “What is this question really asking?” What appears to be a question about data integrity might boil down to something like, “Do you understand the difference between the types of constraints?”

Real World

Orin Thomas

In days past, people ordered products over the telephone by talking to a salesperson. One of the advantages of this was that there was a level of error checking involved at the person-to-person level. With business-to-business communication, if someone isn't fully paying attention, it is possible to order a lot more of something than is actually required. In the past, if someone said to a salesperson, “I want 20,000 rolls of toilet paper,” the salesperson would most likely respond, “Are you sure you need that much?” The sale wouldn't go through until the salesperson was satisfied that 20,000 rolls of toilet paper was exactly what the client wanted.

With the highly automated order systems in place today, unless some sort of error-checking procedure is in place, such an order might simply go through with the appropriate fee automatically charged to the customer. If the client placing the order conducted hundreds of transactions a week, he or she wouldn't know about the error until a semi-trailer truck pulled up with a helpful delivery person asking, “Where would you like us to put this?”

PRACTICE Configuring a Check Constraint

In this practice, you apply a CHECK constraint to the PostalCode column in the Person.Address table of the AdventureWorks database. This CHECK constraint allows users to enter only five-digit numeric postal codes into the column.

1. Open SSMS, and connect to the local instance.
2. Expand the Databases folder.
3. Expand the AdventureWorks database.
4. Expand the Tables folder.
5. Right-click the table Person.Address and choose Modify.
6. Right-click the column PostalCode and choose Check Constraints.
7. Click Add.
8. In the grid, in the Expression field, type **PostalCode LIKE '[0-9][0-9][0-9][0-9]**'.
9. In the Check Existing Data On Creation Or Re-Enabling field, choose No from the drop-down list. This is done because existing table data contains entries that would violate this CHECK constraint.
10. Click Close. On the toolbar, click Save to save your changes to the Person.Address table.

Lesson Summary

- Constraints can be used to enforce data integrity.
- A PRIMARY KEY constraint can encompass multiple columns and enforce uniqueness, but it does not allow for NULL values.
- A UNIQUE constraint applies to a single column and allows a single NULL value.
- A FOREIGN KEY constraint allows only values that exist in the target column. The target column can be in the same table or in a different table, but it must have a PRIMARY KEY or UNIQUE constraint applied.
- A CHECK constraint evaluates entered data against a logical statement.
- A DEFAULT definition sets a default value for a column if none is entered.
- An AFTER trigger performs an action after a modification in a transaction has been made.
- An INSTEAD OF trigger performs an action instead of allowing the modification in the transaction to occur.

Lesson Review

You can use the following questions to test your knowledge of the information in Lesson 2, “Making Implicit Constraints Explicit.” The questions are also available on the companion CD if you prefer to review them in electronic form.

NOTE Answers

Answers to these questions and explanations of why each answer choice is correct or incorrect are located in the “Answers” section at the end of this book.

1. You want to ensure that no contractor can enter more than 60 hours in the Weekly_Hours column of the Worksheet table. Which of the following constraints would you apply?
 - A. UNIQUE
 - B. FOREIGN KEY
 - C. PRIMARY KEY
 - D. CHECK
2. You want to ensure that a value of 0 is entered in the Hours column of the Worksheet table if no other value is specified. Which of the following technologies would you use to achieve this goal?
 - A. Nullability
 - B. CHECK constraint
 - C. DEFAULT definition
 - D. FOREIGN KEY constraint
3. You want to ensure that only names from an approved list of 500 contractors located in the Contractors table can be typed in the Contractor_Name column of the Worksheet table. Which technology will you use to achieve this goal?
 - A. FOREIGN KEY constraint
 - B. UNIQUE constraint
 - C. INSTEAD OF trigger
 - D. AFTER trigger

Lesson 3: Assigning Data Types to Control Characteristics of Data Stored in a Column

The characteristics of how SQL Server stores data within a column, especially numerical data, can be extremely important. In engineering and scientific applications, the precision of a measurement can be critical. There is a big difference in knowing whether a value is accurate to a millimeter or one-tenth of a millimeter. If you do not configure data types correctly, SQL Server might alter data stored in a database to appear less or more precise than it actually is. A column in a table that automatically rounds off a decimal place on entered data might cause serious problems and even require that all measurements be taken again.

After this lesson, you will be able to:

- Assign the Transact-SQL data types to control characteristics of data stored in a column.
- Use Alias data types to ensure that columns share the same length, type, and nullability.

Estimated lesson time: 30 minutes

Transact-SQL Data Types

Sometimes ensuring data integrity can be as simple as ensuring that the appropriate column is assigned one of the existing Transact-SQL data types. If you want to make sure that users can insert only numbers, and not text, in a column used for storing phone numbers, you should make sure that the column uses the int data type rather than the varchar data type. There are 28 base Transact-SQL Data types available in SQL Server 2005. These data types are detailed in Table 8-1.

Table 8-1 Description of Transact-SQL Data Types

Data Type	Description
bigint	Integer between -2^{63} and $(2^{63})-1$.
int	Integer between -2^{31} and $(2^{31})-1$.
smallint	Integer between -2^{15} and $(2^{15})-1$.
tinyint	0 to 255.
bit	0, 1, or NULL.
decimal	A total of 38 digits can be stored on both the left-hand and right-hand sides of the decimal point.

Table 8-1 Description of Transact-SQL Data Types

Data Type	Description
numeric	Functionally equivalent to decimal.
money	−922,337,203,685,477.5808 to 922,337,203,685,477.5807.
smallmoney	−214,748.3648 to 214,748.3647.
float	Floating point numeric data in the ranges of −1.79E+308 to −2.23E−308, 0 and 2.23E−308 to 1.79E+308.
real	Floating point numeric data in the ranges of −3.40E + 38 to −1.18E−38, 0 and 1.18E−38 to 3.40E + 38.
datetime	January 1, 1753 through December 31, 9999. 3.33 millisecond accuracy.
smalldatetime	January 1, 1900 through June 6, 2079. One-minute accuracy.
char	Fixed-length, non-Unicode character data with length of n bytes, where n is a value from 1 to 8000.
varchar	Variable-length, non-Unicode character data with length of n bytes, where n is a value from 1 to 8000.
text	Variable-length, non-Unicode data in the code page of the server with a maximum length of $2^{31}-1$ (2,147,483,647) characters.
nchar	Fixed-length, Unicode character data of n characters. n must be a value from 1 through 4000.
nvarchar	Variable-length Unicode character data. n can be a value from 1 through 4000.
ntext	Variable-length Unicode data with a maximum length of $2^{30}-1$ (1,073,741,823) characters.
binary	Fixed-length binary data with a length of n bytes, where n is a value from 1 through 8000.
varbinary	Variable-length binary data with a length of n bytes, where n is a value from 1 through 8000.

Table 8-1 Description of Transact-SQL Data Types

Data Type	Description
image	Variable-length binary data from 0 through $2^{31}-1$ (2,147,483,647) bytes.
cursor	A data type for variables or stored procedure OUTPUT parameters that contain a reference to a cursor.
sql_variant	A data type that stores values of various data types supported by SQL Server 2005, except text, ntext, image, timestamp, and sql_variant.
table	A special data type that can be used to store a result set for processing at a later time. The table data type is primarily used for temporary storage of a set of rows returned as the result set of a table-valued function.
timestamp	Generally used as a mechanism for version stamping table rows. It is incremented for each insert or update operation performed on a table that contains a timestamp column within a database.
uniqueidentifier	16-byte GUID.
xml	Stores XML data. It is new to SQL Server 2005. Content can be restricted to a well-formed XML fragment or document. It is also possible to specify the XML schema collection.

NOTE Using smallint rather than int or bigint

You should use types such as smallint and tinyint as a way of optimizing data storage within extents. These types reserve less data space and increase database efficiency. You can read more about extents in Chapter 4, "Disaster Recovery."

Alias Data Types

Alias types are based on system data types. You use alias types when several tables within a database must store the same type of data in a column and you need to ensure that these columns have an identical data type, length, and nullability. For example, you might be working with scientific data and want to ensure that all tables

within the database store measurements using the same level of precision. The simplest way of accomplishing this is to create an appropriate alias type and apply it to all columns that must store the scientific measurements within the database.

If you create an alias type in the model database, SQL Server copies it to all new user-defined databases you create. If you create the alias type within a user-defined database, that type exists only within that database. When you create an alias type, you must specify the following parameters:

- Name
- Nullability
- System data type on which the new data type is based

To create a type like the one mentioned in the preceding example called `Sci_Measure`, you would issue the following Transact-SQL statement:

```
CREATE TYPE Sci_Measure
FROM decimal (6,3)
NOT NULL
;
```

MORE INFO **Creating types**

To learn more about creating alias types, you should consult the following MSDN article: [msdn2.microsoft.com/en-us/library/ms175007\(SQL.90,d=ide\).aspx](https://msdn2.microsoft.com/en-us/library/ms175007(SQL.90,d=ide).aspx).

Quick Check

1. What are alias types useful for?
2. Which Transact-SQL data type would you configure a column to use if you needed to store only values 0, 1, or NULL?
3. Which Transact-SQL data type is functionally equivalent to the decimal data type?

Quick Check Answer

1. You use alias types for ensuring that a set of columns are of the same length, type, and nullability.
2. You would use the `bin` data type to accomplish this task.
3. The numeric data type is functionally equivalent to the decimal data type.

Real World

Orin Thomas

We've all heard the story about the guy who figured out how to siphon off all the rounding errors in a bank's daily financial transactions into his own account. This is a great example of why it is important to be as precise as possible when setting up data types. As that guy found, a rounding error of even the smallest fraction of a cent over the course of millions of transactions turns out to be quite a pile of money. In scientific calculations, you can be only as precise as your most imprecise measurement allows. It can also go the other way. If you think that your data is more accurate than it actually is, you might detect things when data-mining that are not actually there. In many real-world applications, you need to record data as accurately as possible.

User-Defined CLR Types

SQL Server 2005 introduces common language runtime (CLR) user-defined types (UDTs). UDTs allow you to extend the scalar type system of the server, and they allow you to store CLR objects within a SQL Server 2005 database. UDTs can contain more than one element. For example, a single UDT could contain numerical and image data. This differentiates UDTs from alias data types, which are restricted to a variation of a base SQL Server data type. UDTs are best suited to storing the following types of data:

- Date, time, currency, and extended numeric types
- Geospatial applications
- Encoded or encrypted data

Creating UDTs within SQL Server 2005 consists of the following stages:

- **Coding and building the assembly that defines the UDT.** UDTs are created using any languages supported by the Microsoft .NET Framework CLR. Data is exposed as fields and properties of a .NET Framework class or structure. Behaviors are defined by methods of the class or structure.
- **Assembly registration.** Use the CREATE ASSEMBLY statement to copy the assembly into the database.
- **Create the UDT in SQL Server.** After you have registered the assembly, use the CREATE TYPE statement to create a UDT.
- **Create tables, variables, or parameters using the UDT.** Use the UDT as a part of the column definition for a table.

Because of their complex nature, UDTs are something that you should be aware of for the exam, but they are not something that you should expect to have to create from scratch in the exam environment.

NOTE Executing CLR types

By default, the ability to execute CLR code within SQL Server 2005 is disabled. You can enable the execution of CLR code by using the *sp_configure* system stored procedure.

BEST PRACTICES Using UDTs

UDTs are accessed by the system as a whole, and using them frequently can degrade performance.

PRACTICE Creating an Alias Data Type

In this practice, you create a new alias data type in the AdventureWorks database and then verify that you can apply the new data type to existing columns if necessary.

1. Use SSMS to connect to the local instance.
2. Expand the Databases folder to view the databases installed on the local instance.
3. Right-click the AdventureWorks database and choose New Query.
4. In the Query window, type the following Transact-SQL statement:

```
CREATE TYPE Sci_Measure
FROM decimal (8,3)
NOT NULL ;
```
5. Click the Execute button on the toolbar.
6. In the Messages pane, ensure that the Command(s) Completed Successfully message is displayed.
7. Expand the AdventureWorks database.
8. Expand the Tables folder.
9. Right-click the Person.Address table and choose Modify.
10. Select the StateProvinceID column, and use the Data Type drop-down list on the Column Properties tab to verify that you can change the column's data type to use Sci_Measure.
11. Return the StateProvinceID column to its original int value, and close the Tables window.

Lesson Summary

- You can use the 28 base Transact-SQL data types to ensure that users can enter only a specific type of data into a column
- An alias data type is a custom version of one of the 28 base Transact-SQL data types.
- Alias data types are often used to standardize the formatting of data across many tables within a database.
- User-defined types (UDTs) allow you to extend the scalar type system, and they allow the storage of CLR objects within a SQL Server 2005 database.
- UDTs can contain more than one element.

Lesson Review

You can use the following questions to test your knowledge of the information in Lesson 3, “Assigning Data Types to Control Characteristics of Data Stored in a Column.” The questions are also available on the companion CD if you prefer to review them in electronic form.

NOTE Answers

Answers to these questions and explanations of why each answer choice is correct or incorrect are located in the “Answers” section at the end of this book.

1. You want to ensure that all recorded measurements in the geological, mineralogical, and paleontological tables are recorded with the same degree of precision. How can you ensure this happens?
 - A. Create an alias data type.
 - B. Use the varchar data type.
 - C. Use the char data type.
 - D. Use the nchar data type.
2. Which of the following types of data would you likely use a UDT to store? (Choose all that apply.)
 - A. Postal codes
 - B. Encrypted data
 - C. Geospatial data
 - D. Birthdays

Chapter Review

To further practice and reinforce the skills you learned in this chapter, you can perform the following tasks:

- Review the chapter summary.
- Review the list of key terms introduced in this chapter.
- Complete the case scenarios. These scenarios set up real-world situations involving the topics of this chapter and ask you to create a solution.
- Complete the suggested practices.
- Take a practice test.

Chapter Summary

- Conflicts occur when updates to the same data occur at different Subscribers at the same time and those Subscribers then synchronize with the Publisher. These conflicts are resolved through the application of conflict resolution policies.
- PRIMARY KEY constraints can encompass multiple columns and enforce uniqueness, but they do not allow for NULL values. Multiple UNIQUE constraints can be applied to a table. FOREIGN KEY constraints allow only values that exist in the target column. CHECK constraints evaluate entered data against a logical statement. DEFAULT definitions set a default value for a column if none is entered.
- An AFTER trigger performs an action once a modification in a transaction has been made. An INSTEAD OF trigger performs an action instead of allowing the modification in the transaction to occur.
- The 28 base Transact-SQL data types can be used to ensure that only a specific type of data can be entered into a column. An alias data type is a custom version of one of these 28 types and is used to standardize the formatting of data across many tables within a database.
- User-defined types (UDTs) allow you to extend the scalar type system, can contain more than one element, and allow the storage of CLR objects.

Key Terms

Do you know what these key terms mean? You can check your answers by looking up the terms in the glossary at the end of the book.

- AFTER trigger
- alias data type
- Article
- CHECK constraint
- conflict resolution policy
- DEFAULT definition
- FOREIGN KEY constraint
- INSTEAD OF trigger
- nullability
- PRIMARY KEY constraint
- Publisher
- Queue Reader Agent
- Subscriber
- Transact-SQL data type
- UNIQUE constraint
- user-defined data type

Case Scenarios

In the following case scenarios, you will apply what you've learned about defining constraints and enforcing data integrity. You can find answers to these questions in the "Answers" section at the end of this book.

Case Scenario 1: Making Implicit Constraints Explicit

Contoso provides IT support for the department of automobile licensing and registration. The department has 20 offices in metropolitan and regional areas. The Central Business District (CBD) office has a SQL Server 2005 computer that acts as a Publisher. Each of the 19 offices has a SQL Server 2005 computer configured as a Subscriber.

1. The department wants to ensure that if a conflict occurs, the SQL Server 2005 computer in the CBD has priority. Which conflict resolution policy should be implemented?
2. The department wants to ensure that only one of the approved localities from the localities table can be entered in the drivers_license table. How might this be achieved?
3. The department wants to ensure that license plate data entered into the database contains three letters followed by three numbers. How might this be achieved?

Case Scenario 2: Data Types

Tailspin Toys is setting up a table within its inventory database to record information about its network. This table will record information such as the Ethernet card address, which software is installed on each computer, and licensing information.

1. Which Transact-SQL data type would be most suitable for storing each number in an IPv4 IP address?
2. Which Transact-SQL data type would be the most suitable for storing the 12-digit hexadecimal MAC addresses of ethernet cards?
3. An inventory utility writes each computer's software configuration to a consistently formatted, extensible markup language file. Which Transact-SQL data type would you apply to a column that you want to use to store these files?

Suggested Practices

To help you successfully master the exam objectives presented in this chapter, complete the following practice tasks:

Design Data Integrity

- **Practice 1: Use a conflict resolution policy.** Create a new database on server Melbourne and configure it as a Publisher. Assign the conflict resolution policy that has Subscriber changes take precedence.
- **Practice 2: Apply check constraints.** In the new database you created on server Melbourne in Practice 1, create a table. Configure a check constraint on a row in this table so that values above 60 will not be accepted.
- **Practice 3: Apply Transact-SQL data types.** Create three new rows in the table that was created in Practice 2. Use Transact-SQL data types to ensure that only the

values 0, 1, or NULL can be entered in the first row; that only dates between January 1, 1900, and June 6, 2079 can be entered in the second row; and that Unicode data of exactly 12 characters can be entered in the third row.

- **Practice 4: Use alias types.** Create an alias data type that ensures that all numbers are no more precise than two digits after the decimal point.

Take a Practice Test

The practice tests on this book's companion CD offer many options. For example, you can test yourself on just one exam objective, or you can test yourself on all the 70-444 certification exam content. You can set up the test so that it closely simulates the experience of taking a certification exam, or you can set it up in study mode so that you can look at the correct answers and explanations after you answer each question.

MORE INFO Practice tests

For details about all the practice test options available, see the "How to Use the Practice Tests" section in this book's Introduction.

Chapter 9

Business Requirements

A database management system, and the databases it contains, has a single primary function—to enable an organization to carry out its business efficiently and effectively. A modern organization, whether commercial, public sector, or charitable, stands or falls by the reliability, quality, and usability of its information. Data needs to be available in a format that users can easily understand and use. Users require information to be as accurate as possible, given that the process of raw data acquisition often results in inaccuracy. Managers not only need to know what the picture is now, they also need analyses, models, and predictions so that they can make forward-looking business decisions. Data from disparate sources needs to be standardized, and methods need to be provided for creating relationships between apparently unrelated datasets.

Configuring, analyzing, and querying data to meet business requirements is a complex process. Fortunately, Microsoft SQL Server 2005 provides powerful tools to assist the database administrator (DBA) in cleaning data, merging incompatible datasets, and creating database models. However, not only do you need to be able to use the tools, you also need to understand the problem at hand.

Exam objectives in this chapter:

- Enforce data quality according to business requirements.
 - Establish the business requirements for quality.
 - Create queries to inspect the data.
 - Use checksum.
 - Clean the data.
- Optimize a database control strategy to meet business requirements.
 - Verify that database change control procedures are being followed.
 - Identify all database objects related to a particular deployment.

Lessons in this chapter:

- Lesson 1: Enforcing Data Quality According to Business Requirements 455
- Lesson 2: Optimizing a Database Change Control Strategy to Meet Business Requirements. 494

Before You Begin

To complete the lessons in this chapter, you must have done the following:

- Configured a Microsoft Windows Server 2003 R2 computer with SQL Server 2005 Enterprise Edition SP1 as detailed in the Appendix.
- Installed an updated copy of the AdventureWorks sample database as detailed in the Appendix.

No additional configuration is required for this chapter.

Real World

Ian McLean

Sometimes configuring a database information system to meet the requirements of users is more about educating users than about configuring systems. User requirements can sometimes verge on expecting the magical. The following story is true—you can't make this sort of thing up.

A senior member of management once approached me, asking for contact information for “the fellow in sales that has the big mustache.” I explained the legal and ethical difficulties of including physical descriptions in a business database—this was before the days when databases could store digital photographs. I don't think the manager was convinced.

A few days later, I saw the same manager shaking hands with one of our salespeople. When the third party left, the manager turned to me and said, “That's the man I wanted your database to find. I remembered his name and found him myself.”

“But,” I said, “he's clean shaven.”

The manager responded, “Oh, he had a mustache when he joined us. He shaved it off years ago.”

Lesson 1: Enforcing Data Quality According to Business Requirements

Establishing and enforcing the business requirements for quality is arguably one of the most difficult aspects of database management. So many variables exist that many people consider it an art rather than a science—and a black art at that. You need to analyze what an organization expects and requires from its data. You need to be aware of the requirements imposed by laws and regulations. You need to consider the quality of the data held in your databases, and the various factors that can affect the quality of that data. You need to be familiar with the tools and techniques that you can use to check, measure, and improve data quality.

In this chapter, we look at business and regulatory requirements and discuss how you can quantify and maintain data quality. We consider methods of analyzing and improving the accuracy and relevance of data, and how you can use data to generate *mining models* and make predictions. Data quality is often determined by the applications that use the data, and we discuss the tools that developers can use to ensure applications enforce data quality. We look at generating queries, eliminating duplicates, and checking data from another location.

NOTE Service packs

The service pack level at the time of this writing is Service Pack 1 (SP1). Unless otherwise indicated, all the information in the chapter applies to both SQL Server 2005 and SQL Server 2005 SP1.

After this lesson, you will be able to:

- Identify business and regulatory requirements that your database implementation needs to satisfy.
- Identify any exceptions to database policy that you need to implement to meet business and regulatory requirements.
- Establish the business requirements for quality, and check that accurate data is recorded.
- Explain how Microsoft SQL Server Integration Services (SSIS), fuzzy algorithms, and data mining clean data and generate models that can predict future trends and help managers make business decisions.
- Create queries to inspect data by using left and right outer joins, full outer joins, and unions and by using the HAVING clause.
- Use binary checksum to check data replicated or copied to another location.
- Clean data by discarding duplicates, storing duplicates for individual inspection, and using the UPDATE statement.

Estimated lesson time: 60 minutes

Analyzing Business and Regulatory Requirements and Determining Exceptions

Typically, a DBA sets up database-system security; designs databases, tables, views, indexes, triggers, constraints, and so on; configures replication; implements a back-up regime; and ends up with a system that is technically excellent but almost unworkable for the typical user. At this point, you need to interview customers and stakeholders to learn their requirements for using the system.

Gathering business and regulatory requirements is a basic step for creating a robust, secure system that provides users with the functionality they require. You need to be able to deal with badly formed queries, *dirty data*, and users who genuinely need additional privileges to do their jobs. At the same time, your system must remain as secure as possible and the validity of data must be maintained. You need to bear in mind the following considerations when gathering information about business and regulatory requirements:

- **Implement a secure and robust first approach.** Before gathering business and regulatory requirements, you should define a restrictive security policy and a sound and robust policy for ensuring data validity, data availability, and disaster recovery. You then incorporate user requirements into the policy as exceptions. Do not attempt to second-guess user requirements when you first design your policies.
- **Interact with business owners and company management.** To gather business requirements, you need to talk to business owners and company management and find out what data and operations are especially sensitive. For example, you might initially decide to encrypt all data that passes through your network, and then find you can limit encryption to a subset of particularly sensitive information.
- **Review regulatory acts.** You need to work with your organization's legal department to review all regulatory acts that could affect your database policy—for example, the California Database Protection Act.
- **Keep exceptions to a minimum.** You must define exceptions to accommodate business and regulatory requirements. If, however, too many exceptions exist, your policies become difficult to manage.
- **Evaluate the risk introduced by exceptions.** For each exception, you need evaluate the risk it introduces in your policies, list all possible methods of implementing the exception, and then review the list to select the best method.

Establishing Business Requirements for Data Quality

Problems with data quality are usually expensive and difficult to eradicate. You have a problem if data in your system does not mean what your users think it does (or should) or if the data does not meet its specification because of mistakes in user entries, errors in transmission, glitches, and so on. Data can also be difficult to understand and categorize because of complexity or lack of *metadata*. Resolving data quality problems is often one of the most time-consuming tasks that a DBA performs.

Traditionally, to meet quality standards, data needs to meet the following criteria:

- **Accuracy and completeness** The data is recorded correctly, and all relevant data is recorded.
- **Uniqueness** Entities are recorded once.
- **Timeliness** The data is kept up to date.
- **Consistency** The data agrees with itself.

However, these criteria require further investigation and qualification. Accuracy and completeness are extremely difficult to measure, and accuracy is not an absolute measurement. If I am looking for a figure for the population of the United States, I would probably be happy with 300 million, whereas if I am balancing a set of accounts, I need data accurate to the dollar, or even to the cent. If you are computing aggregates, you can tolerate a degree of inaccuracy.

The conventional definitions provide no guidance of what is important; how you should define the keys to your data; how you measure interpretability, accessibility, and metadata; and how suitable your data is for analysis. Nor do they provide guidance about practical improvements to the data.

You cannot define data quality in isolation, outside the context of the organization that is using the data. A DBA needs to consult with business analysts, managers, and users to determine the business requirements for data and to generate a definition of data quality that is measurable, reflects the use of the data, and leads to improvements in the processes used. As a starting point, the DBA must have a very clear idea of how to standardize data elements, establish metrics for measuring data quality, and use these metrics to monitor the quality of the data.

The first step toward improving data quality is to formulate a clear understanding of how and where data quality problems occur. Typically, data is not static but instead flows in a data collection and usage process during the following operations:

- Data gathering

- Data delivery
- Data storage
- Data integration
- Data retrieval
- Data mining and analysis

Data Gathering

Data can be transferred or replicated from other systems, or gathered from measuring devices. However, a large proportion of data is entered manually. This can lead to different people using different standards for content and format, to two people entering the same information at the same time, and to approximations and surrogates—you say “Mr.” and I say “Mister.”

You can use built-in integrity checks to tackle data-gathering problems. Sometimes the solution is managerial—for example, rewarding staff for accurate data entry can bring about genuine improvements. Techniques such as duplicate removal and field value standardization, and the use of data validation and fuzzy logic, can help address the problems associated with badly entered data.

Data Delivery

When data is delivered to your system, inappropriate pre-processing can corrupt or destroy it. Inappropriate aggregation, nulls converted to default values, buffer overflows, and transmission glitches can cause problems.

You need to verify data by using data verification, checksums, and verification parsers. Data verification can check whether uploaded files fit an expected pattern and whether dependencies exist between data streams and processing steps. If your organization has a datastream supplier, this supplier should have made a formal commitment to data quality.

Data Storage

Problems in physical storage can exist, but disk storage is relatively inexpensive, and you can solve most physical problems by installing faster or larger disks and disk arrays that provide failover protection. You should also formulate and enforce a sound backup and restore policy, and place your transaction files on a separate disk from your database files. Data storage problems more typically arise from poor metadata

that causes difficulty in identifying and retrieving data, or from data feeds generated by legacy applications resulting in inappropriate data models. Missing timestamps, incorrect normalization, and ad-hoc modifications can also cause data storage problems.

You (or possibly the database or application designer) need to ensure that metadata is sufficient, useful, and coherent. Your organization needs to document and publish data specifications. You can use data browsing and data mining to examine the data and check that it meets a written specification.

Data Integration

Combining data sets that your organization might have acquired from elsewhere, or combining data sets across departments, can be a source of problems. Such data sets could have no common keys and use different field formats. At the extreme, you could be combining two columns called “Receipts,” one of which is in dollars and the other in euros. Often the data relates to differing time periods with incompatible time windows.

A significant body of research exists in data integration. However, a detailed discussion of this topic is beyond the scope of this book and the 70-444 examination. Commercial tools are available for address matching, schema mapping, data browsing, and exploration.

Data Retrieval

Exported datasets are often a view of the actual data. Problems can occur because source data and the need for derived data are not properly understood. Often the solution is to look for errors in interpretation—possibly an outer join was used instead of an inner join, or vice versa, or NULL values were not handled correctly.

Data Mining and Analysis

Data mining and analysis poses and hopefully answers the question of what you are actually doing with the data. You need to carefully specify the scale of data mining and the degree of confidence that you can expect from the results. Analysts and statisticians can sometimes be over-attached to data models and lose sight of the overall data domain—or what the data is for in the first place. You need to determine which models and techniques are appropriate. Analysis is a continuous process and should form part of a feedback loop. The section “Using Data Mining” later in this lesson deals with this topic in greater depth.

Quick Check

- You need to determine data quality standards. You need to consult with the business analysts and DBAs in your organization to generate a definition of data quality that is measurable, reflects the use of the data, and leads to improvements in the processes used. What do you need to do to achieve this goal? Give three answers.

Quick Check Answer

- Establish metrics for measuring data quality.
- Identify how to standardize data elements.
- Identify how to monitor the quality of your data.

Ensuring Applications Enforce Data Quality

If you are working with developers to design applications that run against your organization's databases, or if you are specifying commercially available applications, you need to ensure that these applications enforce data quality. You need to determine, for example, how an application manages security requirements; the privileges each user needs to run and benefit from the application; how the application deals with badly entered or dirty data and data from disparate sources (including non-Microsoft, non-SQL Server 2005, and earlier databases); and how well the application provides customers, partners, and employees with the information and data models they need to make better business decisions and to make predictions based on the best available information.

Microsoft provides an application infrastructure and tools—such as SQL Server 2005, Microsoft Visual Studio 2005, and Microsoft BizTalk Server 2006—to enable application developers to build connected systems and applications that meet modern requirements for the quality, cohesiveness, availability, and security of data.

Visual Studio 2005

Visual Studio 2005, shown in Figure 9-1, includes tools to create an application life cycle suite that enables developers, database architects, system administrators, and DBAs to work together. Microsoft has integrated these tools with SQL Server 2005 to provide increased productivity across connected systems—from front-end to database development. Visual Studio 2005 offers the Visual Studio Professional and Video Web Development packages, which provide productive developer tools for all levels of experience, from the novice programmer to the experienced enterprise development team.

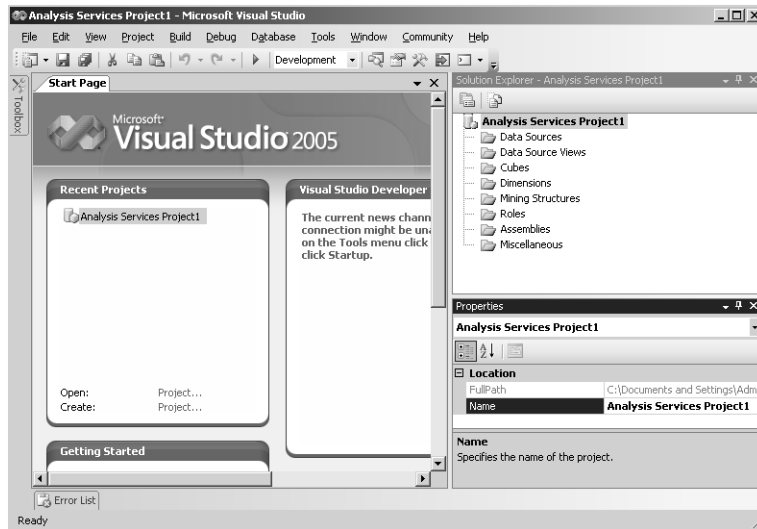


Figure 9-1 Visual Studio 2005.

Visual Studio Professional provides integrated development environment (IDE) enhancements with productivity features such as Snaplines, the Class Designer, SmartTags, and Edit and Continue. It also provides enhancements to programming languages such as Microsoft Visual Basic, Visual C#, Visual C++, and Visual J#, and it enhances the .NET Framework and ASP.NET environments. Visual Studio Professional provides Smart Client Development technologies, including Windows Form development, .NET Compact Framework development, and Visual Studio Tools for Microsoft Office development. Video Web Development provides enhanced facilities for Web site developers and is outside the scope of this chapter.

MORE INFO Visual Studio Home

For more information about Visual Studio, including a guided tour and an opportunity to evaluate the product, access Visual Studio Home at msdn.microsoft.com/vstudio.

BizTalk Server 2006

BizTalk Server 2006 is a business process management (BPM) system that enables companies to automate and optimize business processes. It provides tools that enable development teams to design, deploy, and manage those processes.

BizTalk Server 2006 formalizes the concept of a BizTalk application by providing a logical container for housing all the components (artifacts) for a given solution. This

enables administrators to work with a complete BizTalk application as a unit and simplifies the management, troubleshooting, and deployment of business processes. A developer can use the BizTalk Administration Console to package applications into Windows Installer (.msi) files, greatly simplifying installation.

BizTalk Server 2006 consolidates all management functionality into the BizTalk Administration Console. Users can use the console to create artifacts and messaging components in addition to configuring, deploying, stopping, and starting applications across multiple servers. An administrator can use the BizTalk Administration Console's Group Hub page to monitor the health of currently running BizTalk applications. The Group Hub page uses color-coded indicators to display problems.

The Business Activity Monitor (BAM) portal is enhanced to allow an information worker to easily examine and configure BAM information. The user can select a particular instance of a business process to monitor and then choose different BAM views to get different perspectives on the key performance indicators.

MORE INFO BizTalk Server 2006

For more information about BizTalk Server 2006, access www.microsoft.com/biztalk/default.mspx.

Exam Tip You need to be aware of Visual Studio 2005 and BizTalk Server 2006 and to know what their functions are. However, the packages are primarily tools for developers rather than DBAs, and the 70-444 examination is unlikely to test them in any depth.

SQL Server 2005 Business Intelligence Tools

SQL Server 2005 provides a variety of Business Intelligence (BI) tools that assist DBAs, database developers, and application developers—working in a team with systems and business analysts—to ensure that business applications meet the required standards for data quality and usability. This book discusses each of these tools, many in considerable depth, in other chapters, so this section just summarizes their use. Table 9-1 lists various business requirements and the BI tools that SQL Server 2005 provides to implement them.

Table 9-1 Business Requirements and SQL Server 2005 BI Tools

Business Requirement	SQL Server 2005 BI Tools
Extracting, transforming, and loading data	SQL Server Integration Services
Relational data warehousing	SQL Server relational databases

Table 9-1 Business Requirements and SQL Server 2005 BI Tools

Business Requirement	SQL Server 2005 BI Tools
Multidimensional database implementation	SQL Server Analysis Services
Data mining	SQL Server Analysis Services
Managed reporting	SQL Server Reporting Services
Ad hoc reporting	SQL Server Reporting Services
Ad hoc querying and analysis	SQL Server Query Editor. Microsoft Office products (Excel, Office Web Components, Data Analyzer, Share-Point Portal)
Database development tools	SQL Server Business Intelligence Development Studio
Database management tools	SQL Server Management Studio

SQL Server Management Studio (SSMS) and SQL Server Business Intelligence Development Studio (BIDS) are new in SQL Server 2005. SQL Server Integration Services (SSIS), SQL Server Analysis Services (SSAS), and SQL Server Reporting Services (SSRS) are substantially improved. You could argue that SSIS is new in SQL Server 2005, but it is in effect a substantial redesign of the SQL Server 2000 Data Transformation Services (DTS). The SQL Server 2005 relational database contains several significant new features. Although the Office query and portal tools are not part of SQL Server, the current releases work with SQL Server 2005. The BI functionality in Office will evolve in future Office product release cycles.

Microsoft states that a key goal of the SQL Server 2005 BI components is to support the development and use of BI in enterprises of all sizes, and to all employees—not just management and analysts, but also operational and external constituents. The SQL Server 2005 BI toolset integrates application development and ensures that applications enforce data quality by implementing the following functions:

- **Design** BIDS is an integrated development environment designed for the business intelligence system developer. BIDS offers debugging, source control, and script and code development functions for all components of a BI application.
- **Integration** Microsoft has rewritten DTS and renamed it SSIS. You can use SSIS to perform complex data integration, transformation, and synthesis at high

speed for very large amounts of data. SSIS, SSAS, and SSRS work together to present a seamless view of data from heterogeneous sources.

- **Storage** SQL Server 2005 blurs the distinction between relational and multidimensional databases. You can store data in the relational database or in the multidimensional database. You can also use the new proactive cache feature to get the best of both worlds.

MORE INFO Proactive cache

The proactive cache transparently synchronizes and maintains an updated copy of the data organized specifically for high-speed querying and for preventing end users from overloading the relational databases. The structure of the cache is automatically derived from the Universal Data Model (UDM) structure and can be finely tuned to balance performance with latency of data. For more information about configuring storage and proactive caching settings, search for "Configuring Storage" in Books Online or access [msdn2.microsoft.com/en-us/library/ms174784\(d=ide\).aspx](http://msdn2.microsoft.com/en-us/library/ms174784(d=ide).aspx).

- **Analysis** Microsoft has added new data mining algorithms in SQL Server 2005, including Association Rules, Time Series, Regression Trees, Sequence Clustering, Neural Nets, and Naïve Bayes. New analytical capabilities are also added to SSAS cubes, including the Key Performance Indicator framework and Multidimensional Expressions (MDX) scripts. The SSRS report delivery and management framework enables easy distribution of complex analytics to the widest possible audience. Detailed discussion of these enhanced analysis features is beyond the scope of this book.
- **Reporting** SSRS provides an enterprise managed reporting environment, embedded and managed by using Web services. Reports can be personalized and delivered in a variety of formats and with a range of interactivity and printing options. Complex analyses can reach a broad audience through the distribution of reports. Report Builder, new in SQL Server 2005, provides for ad hoc reporting by the end user.
- **Management** SSMS integrates the management of all SQL Server 2005 components. Through SSMS, BI platform components gain enhanced scalability, reliability, availability, and programmability. These enhancements provide significant benefits to the BI practitioner.

SQL Server 2005 provides different tools for different jobs. You need to become familiar with them and know what they do. Do not try to make one do the work of another. For example, in an environment such as a data warehouse where much more data is coming out of the system than is being put into it, you typically use the functions that SSRS provides. You can use SSAS for putting together multidimensional data analysis

structures such as cubes, but such structures typically benefit from intelligent optimization, the success of which often depends on how well the data itself is designed.

Using SQL Server Integration Services

SQL Server 2005 introduces SSIS, which replaces (or rather substantially upgrades) DTS in SQL Server 2000. You can use SSIS packages to merge data from heterogeneous data sources, populate data warehouses, clean and standardize data, build BI into a data transformation process, and automate administrative functions.

Business Intelligence Development Studio

You can use BIDS to design and construct SSIS packages. You can also use BIDS to secure sensitive data stored in packages, troubleshoot package execution by using checkpoints, and deploy completed SSIS packages to other servers. BIDS is not restricted to SSIS packages. You can also use it to develop SSAS and SSRS projects.

As shown in Figure 9-2, BIDS provides the following panes:

- **The Toolbox pane** Contains items that can be used in constructing projects. The availability of items is dependent on the current task.
- **The SSIS Designer pane** Used to create or modify business intelligence objects.
- **The Solution Explorer pane** Contains all items associated with the current project.
- **The Properties pane** Contains the properties of an object.

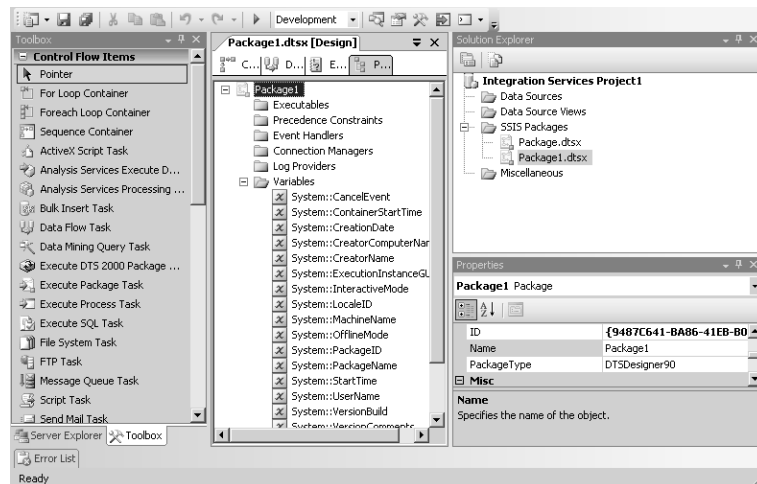


Figure 9-2 BIDS panes.

BIDS Toolbox

The BIDS toolbox is a collection of tasks that you can drag from the Toolbox to the Designer window as you build your project. The Toolbox contains collections of tasks (sections) that are relevant to your current tasks. For example, when you are editing control flow, the Control Flow Items and Maintenance Plan Tasks sections are available in the Toolbox. When you are editing data flow, the Data Flow Sources, Data Flow Transformations, and the Data Flow Destinations sections are available in the Toolbox.

SSIS Designer Window

SSIS Designer is a component of BIDS that you can use to construct the control flow in a package, add event handlers to the package and package objects, view package content, and view the execution progress of a package. When a package executes, an additional tab appears displaying progress and execution results.

SSIS Packages

A package is an organized collection of tasks and workflow elements. The order of task execution depends on the outcome of earlier steps in the workflow. Packages can be constructed visually by using BIDS or programmatically by using (for example) Visual Studio 2005.

Control Flow and Data Flow

A control flow task generally operates on files that contain data—for example, copying a file or executing a script. A data flow task operates on the data itself—for example, extracting data from a database table and sorting it.

To include data flow tasks in your SSIS package, you must drag a data flow task onto the control flow pane and select it. When you have selected the data flow task, you need to navigate to the data flow pane to configure the data flow for that task. If you have multiple data flow tasks on the control flow pane, the information displayed on the data flow pane changes depending on which data flow task is selected on the control flow pane.

Data Flow Pipeline Components

The data flow engine supports data flow tasks that are dedicated to moving and transforming data from disparate sources. Data flow tasks contain objects called *pipeline components*, which define the movement and transformation of data. Programming the data flow engine lets developers automate the creation and configuration of the components in a data flow task and create custom components.

MORE INFO SQL Server:SSISPipeline object

If the execution time of SSIS packages increases suddenly and dramatically, and you suspect that this might be because of memory problems, you should monitor all the counters associated with the *SQL Server:SSIS Pipeline* performance object. For more information, search for "Monitoring Performance of the Data Flow Engine" in Books Online.

Deploying a SSIS Package

When you have created, tested, and debugged your SSIS package, you then deploy it on your production database. If you want to deploy an SSIS package, you first need to configure a build process that creates a deployment utility for an SSIS project. The deployment utility is a folder that contains the files you need to deploy the packages in an SSIS project on a different server. You create the utility on the computer on which the SSIS project is stored.

When you build the project, all packages and package configurations in the project are automatically included in the utility. To deploy additional files with the project, place the files in the Miscellaneous folder of the SSIS project. When the project builds, these files are also automatically included.

You can configure each project deployment differently. Before you build the project and create the package deployment utility, you can set the properties on the deployment utility to customize the way the utility deploys packages in the project. For example, you can specify whether package configurations can be updated during deployment.

When you build an SSIS project, BIDS creates a manifest file and adds it, together with copies of the project packages and package dependencies, to the `bin\Deployment` folder in the project or to the location specified in the *DeploymentOutputPath* property. The manifest file lists the packages, the package configurations, and any miscellaneous files in the project.

You can deploy SQL Server 2005 Integration Services (SSIS) packages by using a deployment utility, by using the import and export package features in SSMS, or by saving a copy of the package in BIDS. However, only the deployment utility can deploy multiple packages, including the package dependencies (for example, configurations and table names) and the files that contain supporting information (for example, documentation).

For example, if you create an SSIS project on your test network and want to deploy it on your production network, but your production network uses different table names

than your test network, you can copy the deployment folder for your SSIS project to your production servers and then execute the manifest file to change the table names as required.

To create a package deployment utility, open the solution that contains the SSIS project for which you want to create a package deployment in BIDS, right-click the project, and then choose Properties. In the Property Pages dialog box, select Deployment Utility as shown in Figure 9-3.

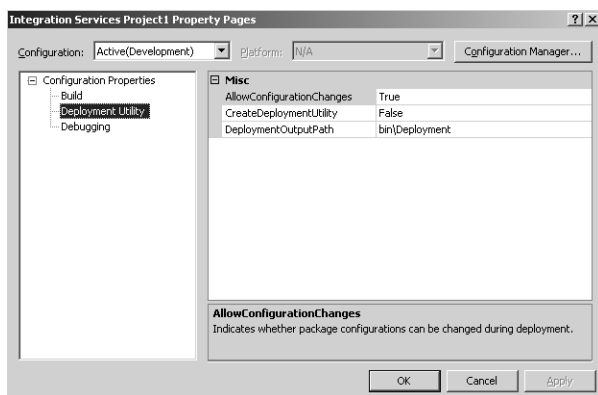


Figure 9-3 SSIS Deployment Utility.

You can then configure the *AllowConfigurationChanges* and *CreateDeploymentUtility* properties and, optionally, update the location of the deployment utility by modifying the *DeploymentOutputPath* property. You then click OK, and in Solution Explorer you right-click the project and choose Build.

When you have built the deployment utility, you can copy the collection of files to your target servers. To install the package on the target servers, you need to run the Package Installation Wizard. When you have verified that your SSIS package has installed correctly, you can use Microsoft SQL Server Agent to configure a schedule for package execution.

Using Fuzzy Transformations

Real-world data is dirty because of misspellings, truncations, missing or inserted tokens, null fields, unexpected abbreviations, and other irregularities. During the extract, transform, and load (ETL) phase, SQL Server 2005 cleans and standardizes new data and makes it consistent with existing data. The Fuzzy Lookup and Fuzzy Grouping transformations make the ETL process more resilient to common data

errors. They address matching and grouping problems without requiring complex, domain-specific rules and scripts.

IMPORTANT Fuzzy Grouping and Fuzzy Lookup transformations

Fuzzy Grouping and Fuzzy Lookup transformations are available only in the Enterprise version of SQL Server 2005.

Fuzzy Lookup lets you match records that users enter with clean, standardized records in a reference table. The matching process takes account of errors that are present in the input records. Fuzzy Lookup returns the closest match and indicates the quality of that match. Suppose, for example, customer information that a sales assistant enters during a new sales transaction does not match exactly with any record in the Customers reference table, which consists of all current customers, because of typographical or other errors in the input data. Fuzzy Lookup returns the best matching record from the Customers reference table and provides measures to indicate the match quality.

Fuzzy Grouping enables you to identify groups of records in a table where each group potentially corresponds to the same real-world entity. Records in each group might not be identical but are very similar to each other. Fuzzy Grouping is used, for example, to group together all records in a Customers reference table that describe a single real customer.

Fuzzy Lookup and Fuzzy Grouping use a custom, domain-independent *distance function* that takes into account the edit distance (for example, “hits” is distance 2 from “bit”), the number of tokens, the token order, and relative frequencies. As a result, Fuzzy Lookup and Fuzzy Grouping achieve much finer discrimination than full-text searches because they capture a more detailed structure of the data. You can use Fuzzy Lookup and Fuzzy Grouping with little or no custom programming for ETL tasks in SQL Server 2005.

Because they use more than just edit distance, Fuzzy Lookup and Fuzzy Grouping are not easily misled by transpositions and can detect higher level patterns than an approach that uses only edit distance.

The Fuzzy Lookup Transformation

Fuzzy Lookup can find data in large tables by using an incorrectly entered or incomplete string key. For example, if you want to look up customer information by name and address, you can use Fuzzy Lookup to find the information, even if your input does

not match exactly what is stored in your reference table. The Fuzzy Lookup transformation performs data cleaning tasks such as standardizing data, correcting data, and providing missing values.

NOTE Fuzzy Lookup and Lookup transformations

A Fuzzy Lookup transformation frequently follows a Lookup transformation in a package data flow. First, the Lookup transformation tries to find an exact match. If it fails, the Fuzzy Lookup transformation provides close matches from the reference table.

The transformation requires access to a reference data source that contains values used to clean and extend the input data. The reference data source must be a table in a SQL Server 2005 database. The match between the value in an input column and the value in the reference table can be an exact match or a fuzzy match. However, the transformation requires at least one column match to be configured for fuzzy matching. You can customize this transformation by specifying the maximum amount of memory, the row comparison algorithm, and the caching of indexes and reference tables that the transformation uses.

The Fuzzy Lookup transformation includes three features for customizing the lookup it performs: maximum number of matches to return per input row, token delimiters, and similarity thresholds. The transformation provides a default set of delimiters used to tokenize the data, but you can add token delimiters to suit the needs of your data. The Delimiters property contains the default delimiters. Tokenization is important because it defines the units within the data that are compared to each other.

The similarity thresholds can be set at the component and join levels. The join-level similarity threshold is available only when the transformation performs a fuzzy match between columns in the input and the reference table. You specify the similarity threshold by setting the *MinSimilarity* property at the component and join levels.

Each match includes a similarity score and a confidence score. The similarity score is represented by a value between 0 and 1, where 1 means an exact match between the value in the input column and the value in the reference table. The confidence score, also a value between 0 and 1, indicates the confidence in the match. If no usable match is found, similarity and confidence scores of 0 are assigned to the row, and the output columns copied from the reference table will contain null values.

The Fuzzy Grouping Transformation

The Fuzzy Grouping transformation performs data-cleaning tasks by identifying rows of data that are likely to be duplicates and selecting a row of data (the canonical row)

to use in standardizing the data. The transformation requires a connection to an instance of SQL Server 2005 to create the temporary SQL Server tables that the transformation algorithm requires to do its work. The connection must run under the account of a user who has permission to create tables in the database.

To configure the transformation, you must select the input columns to use when identifying duplicates, and you must select the type of match—fuzzy or exact—for each column. An exact match guarantees that only rows that have identical values in that column will be grouped. A fuzzy match groups rows that have approximately the same values. The method for approximate matching of data is based on a user-specified similarity score.

The transformation output includes all input columns, one or more columns with standardized data, and a column that contains the similarity score. The score is a decimal value between 0 and 1. The canonical row has a score of 1. Other rows in the fuzzy group have scores that indicate how well the row matches the canonical row. The closer the score is to 1, the more closely the row matches the canonical row. The transformation does not remove duplicate rows, but instead groups them by creating a key that relates the canonical row to similar rows.

The Fuzzy Grouping transformation includes two features for customizing the grouping it performs: token delimiters and similarity threshold. The transformation provides a default set of delimiters used to tokenize the data, but you can add new delimiters that improve the tokenization of your data.

The similarity threshold indicates how strictly the transformation identifies duplicates. You specify the similarity threshold among rows and columns by setting the *MinSimilarity* property at the component and column levels. To satisfy the similarity that is specified at the component level, all rows must have a similarity across all columns that is greater than or equal to the similarity threshold that is specified at the component level.

To identify a similarity threshold that works for your data, you might have to apply the Fuzzy Grouping transformation several times using different minimum similarity thresholds. At run time, the score columns in transformation output contain the similarity scores for each row in a group. You can use these values to identify the similarity threshold that is appropriate for your data. If you want to increase similarity, you should set *MinSimilarity* to a value larger than the value in the score columns.

Using Data Mining

Microsoft defines data mining as the process of extracting valid, authentic, and actionable information from large databases. Data mining provides business information by deriving patterns and trends that exist in data. These patterns and trends can be collected together and defined as a mining model. For example, mining models can be applied to the following business operations:

- Sales forecasting
- Targeted mailing
- Product placement

Building a mining model is part of a larger process that starts with defining the basic problem that the model will solve, and ends in the deployment of the model in a working environment. This process can be defined by using the following steps:

- Defining the problem
- Preparing data
- Exploring data
- Building models
- Exploring and validating models
- Deploying and updating models

Each step does not necessarily lead directly to the next step. Creating a data mining model is a dynamic and iterative process. After you explore the data, you might find that it is insufficient to create the appropriate mining models, and you then need to look for more data. You might build several models and realize that they do not answer the problem posed when you defined that problem. Therefore, you need to redefine the problem. You might need to update the models after they have been deployed because more data has become available.

Defining the Problem

This step includes analyzing business requirements, defining the scope of the problem, defining the metrics by which the model will be evaluated, and defining the final objective for the project. You need to define what you are looking for, what predictions you want to make, what types of relationships you want to find, whether you want to make predictions or merely look for patterns and associations, how the data is distributed, and how the database columns or database tables are related.

You might need to conduct a data availability study and investigate the needs of business users with regard to the available data. If the data does not support the needs of the users, you might need to redefine the project.

Preparing Data

This step involves consolidating and cleaning the data that was identified when you defined the problem. SSIS contains all the tools that you need to complete this step, including transforms to automate data cleaning and consolidation. Figure 9-4 shows the Data Mining Queries Task Properties screen in SSIS.

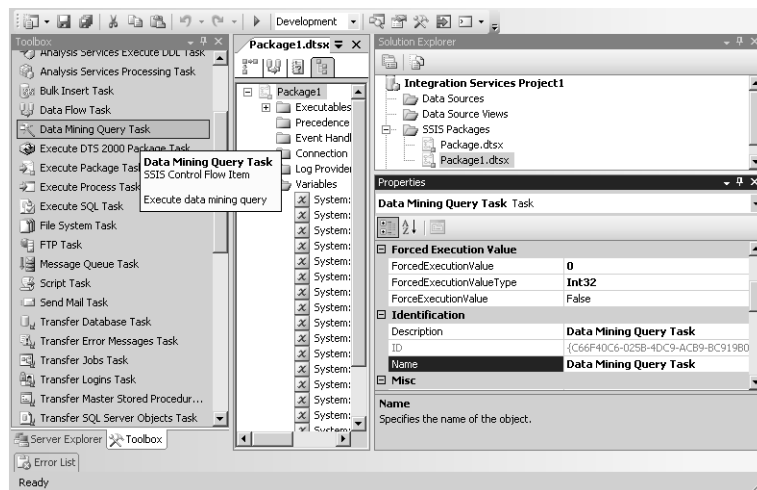


Figure 9-4 Data Mining Queries Task Properties screen.

Data can be scattered across an organization and stored in different formats. It might contain inconsistencies such as flawed or missing entries. For example, the data might show that a customer shops regularly at a store located 1,000 miles from her home. Before you start to build models, you must fix these problems. Typically, you need to use tools such as those provided by SSIS to explore the data and find the inconsistencies.

Exploring Data

You must know enough about the data to make appropriate decisions when you create the models. Exploration techniques include calculating the minimum and maximum values, calculating mean and standard deviations, and looking at the distribution of the data. After you explore the data, you can determine whether

the dataset contains flawed data, and devise a strategy for fixing problems. Data Source View Designer in BIDS contains several tools that you can use to explore data.

Building Models

Before you build a model, you must place the prepared data into separate training and testing datasets. You use the training dataset to build the model, and the testing dataset to test the accuracy of the model by creating prediction queries. You can use the Percentage Sampling Transformation in SSIS to split the dataset.

A mining model is defined by a data mining structure object, a data mining data model object, and a data mining algorithm. It typically contains input columns, an identifying column, and a predictable column. You define these columns by using the Data Mining Extensions (DMX) language or the Data Mining Wizard in BIDS. After you define the structure of the mining model, you then process (or *train*) it by populating the empty structure with the patterns that describe the model. You can find patterns by passing the original data through a mathematical algorithm. SQL Server 2005 contains a different algorithm for each type of model you can build. You can use parameters to adjust each algorithm.

MORE INFO Mining structures and data mining algorithms

For more information, search for "Mining Structures" and "Data Mining Algorithms" in Books Online or access [msdn2.microsoft.com/en-us/library/ms174757\(d=ide\).aspx](http://msdn2.microsoft.com/en-us/library/ms174757(d=ide).aspx) and [msdn2.microsoft.com/en-us/library/ms175595\(d=ide\).aspx](http://msdn2.microsoft.com/en-us/library/ms175595(d=ide).aspx), respectively.

Exploring and Validating Models

Your next step is to explore the models that you have built and test their effectiveness before deploying them in a production environment. You might create several models and then need to decide which will perform best. If none of the models perform well, you need to revisit a previous step in the process, either redefining the problem or reinvestigating the data in the original dataset.

You can use Data Mining Designer in BIDS to explore trends and patterns that the algorithms discover. You can also test how well the models create predictions by using the lift chart and classification matrix tools in the designer.

After you deploy the mining models in a production environment, you can use them to make predictions, which your organization can use to make business decisions.

Application designers can include Analysis Management Objects (AMOs) to create, alter, process, and delete mining structures and mining models. You, or more probably a database designer, can use SSIS to create a package in which a mining model is used to intelligently separate incoming data into multiple tables. You can create a report that lets users directly query against an existing mining model.

Updating the model is part of the deployment strategy. As more data comes into your organization, you should reprocess the models and improve their effectiveness.

Creating Queries to Inspect the Data

In general, when you run queries against a database, your purpose is to identify only the data that meets a particular set of criteria. You want to see the data that is of use to you and nothing else. In this situation, if you were joining two database tables that had a common column, you would typically use an inner join. A query that uses an inner join returns rows only when at least one row from both tables matches the join condition.

Consider, for example, an inner join of the Product and ProductReview tables in the AdventureWorks database on their ProductID columns. The results show only the products for which reviews have been written. For a normal query, this probably is the required result. However, for queries that expose data inconsistencies, you want to see all table entries. In this case, you would design your query to use an outer join.

Using Left and Right Outer Joins

Outer joins return all rows from at least one of the tables or views mentioned in the FROM clause, provided that those rows meet any WHERE or HAVING search conditions. A query retrieves all rows from the left table referenced in a left outer join, all rows from the right table referenced in a right outer join, and all rows from both tables in a full outer join.

SQL Server 2005 uses the following SQL-92 keywords for outer joins specified in a FROM clause:

- LEFT OUTER JOIN or LEFT JOIN
- RIGHT OUTER JOIN or RIGHT JOIN
- FULL OUTER JOIN or FULL JOIN

MORE INFO SQL-92

For more information about the SQL-92 standard, access www.service-architecture.com/database/articles/sql-92.html.

To include all products, regardless of whether a review has been written, you can use a SQL-92 left outer join. The following Transact-SQL query performs this function:

```
USE AdventureWorks;
GO
SELECT p.Name, pr.ProductReviewID
FROM Production.Product p
LEFT OUTER JOIN Production.ProductReview pr
ON p.ProductID = pr.ProductID;
GO
```

The LEFT OUTER JOIN includes all rows in the Product table in the results, regardless of whether there is a match on the ProductID column in the ProductReview table. Notice that in the results where the product review ID for a product has no match, the row contains a null value in the ProductReviewID column.

Consider an inner join of the AdventureWorks SalesTerritory table and SalesPerson table on their TerritoryID columns. The results show any territory that has been assigned to a sales person. However, The SQL-92 right outer join operator RIGHT OUTER JOIN indicates all rows in the second table are to be included in the results, regardless of whether there is matching data in the first table. Thus, if you want to include all sales people in the results, regardless of whether they are assigned a territory, you use a SQL-92 right outer join, as demonstrated in the following Transact-SQL query:

```
USE AdventureWorks;
GO
SELECT st.Name AS Territory, sp.SalesPersonID
FROM Sales.SalesTerritory st
RIGHT OUTER JOIN Sales.SalesPerson sp
ON st.TerritoryID = sp.TerritoryID;
GO
```

You can further restrict an outer join by using a predicate. The following example contains the same right outer join as the previous one, but includes only sales territories with sales less than \$2,000,000:

```
USE AdventureWorks;
GO
SELECT st.Name AS Territory, sp.SalesPersonID
FROM Sales.SalesTerritory st
```



```
RIGHT OUTER JOIN Sales.SalesPerson sp
ON st.TerritoryID = sp.TerritoryID
WHERE st.SalesYTD < $2000000;
GO
```

MORE INFO Predicates

For more information about predicates, search for “Search Condition (Transact-SQL)” in Books Online or access [msdn2.microsoft.com/en-us/library/ms173545\(d=ide\).aspx](https://msdn2.microsoft.com/en-us/library/ms173545(d=ide).aspx).

Using Full Outer Joins

To retain the nonmatching information by including nonmatching rows in the results of a join, use a full outer join. SQL Server 2005 provides the full outer join operator, FULL OUTER JOIN, which includes all rows from both tables, regardless of whether the other table has a matching value.

Consider a join of the AdventureWorks Product table and the SalesOrderDetail table on their ProductID columns. The results show only the Products that have sales orders. The SQL-92 FULL OUTER JOIN operator indicates that all rows from both tables are to be included in the results, regardless of whether there is matching data in the tables.

You can include a WHERE clause with a full outer join to return only rows that have no matching data between the tables. The following query returns only products that have no matching sales orders, as well as sales orders that are not matched to a product (although in this case all sales orders are matched to a product):

```
USE AdventureWorks;
GO
SELECT p.Name, sod.SalesOrderID
FROM Production.Product p
FULL OUTER JOIN Sales.SalesOrderDetail sod
ON p.ProductID = sod.ProductID
WHERE p.ProductID IS NULL
OR sod.ProductID IS NULL
ORDER BY p.Name;
GO
```

If, for example, you were writing a stored procedure that displayed all the items in two tables, both with a product_id column, and you needed to display all products that did not have a valid product ID, you would design the procedure to use an outer join to join two tables by using a product_id column. You would add a WHERE clause to remove all valid product IDs.

MORE INFO The NOT EXISTS subquery

You can use full outer joins with the WHERE clause to list column entries that do not have a specific property—for example, items in a Production database table that do not have a valid product code. An alternative method of obtaining the same result set is to use a query with the NOT EXISTS subquery in the WHERE clause, typically in a stored procedure. For more information, search for “Subqueries with NOT EXISTS” in Books Online or access [msdn2.microsoft.com/en-us/library/ms184297\(d=ide\).aspx](https://msdn2.microsoft.com/en-us/library/ms184297(d=ide).aspx).

Using Cross Joins

Although it is (arguably) less useful than outer joins for inspecting data, you can use the cross join if you need to display the product of two tables. A cross join that does not have a WHERE clause produces the Cartesian product of the tables involved in the join. The size of a Cartesian product result set is the number of rows in the first table multiplied by the number of rows in the second table. The following is an example of a Transact-SQL cross join:

```
USE AdventureWorks;
GO
SELECT e.EmployeeID, d.Name AS Department
FROM HumanResources.Employee e
CROSS JOIN HumanResources.Department d
ORDER BY e.EmployeeID, d.Name;
GO
```

The result set contains 4640 rows because Employee has 290 rows and Department has 16.

If a WHERE clause is used, a cross join behaves as an inner join.

MORE INFO Hash joins

You can use hash joins to tune your queries. You can implement hash joins in many types of set-matching operations—for example: inner join; left, right, and full outer join; left and right semi-join; intersection; union; and difference. You can also use the hash join for duplicate removal and grouping. For more information about hash joins, search for “Understanding Hash Joins” in Books Online or access [msdn2.microsoft.com/en-us/library/ms189313\(d=ide\).aspx](https://msdn2.microsoft.com/en-us/library/ms189313(d=ide).aspx).

Using the UNION Operator

You can combine the results of two or more queries into one result set by using the UNION operator in the Transact-SQL SELECT statement. UNION generates a result set that includes all the rows that belong to all queries in the union. The UNION

operator differs from joins in that it combines query results, whereas joins combine columns from two tables.

If you want to combine the result sets of two or more queries by using UNION, the number and the order of the columns must be the same in all queries and the data types must be compatible. You will find an example of combining the results of two queries by using UNION in the “Practice” section of this lesson.

Using the HAVING Clause

The HAVING clause, used in the Transact-SQL SELECT statement, specifies a search condition for a group or an aggregate (the WHERE clause cannot use aggregates). Typically, you use HAVING in a GROUP BY clause. If you do not use a GROUP BY clause, HAVING behaves like a WHERE clause. You can use HAVING to extract results that contain a specified value from a column. The following example lists all the items in the Sales.SalesOrderDetail table in the AdventureWorks database that have an average order quantity of three or fewer:

```
USE AdventureWorks;
GO
SELECT ProductID
FROM Sales.SalesOrderDetail
GROUP BY ProductID
HAVING AVG(OrderQty) > 3
ORDER BY ProductID;
GO
```

NOTE Finding duplicates by using HAVING

You can write a query that includes the GROUP BY and HAVING clauses to remove duplicate values from its results. You can also use the WHERE clause but, unlike HAVING, WHERE cannot use aggregates. If you want to create a new table that contains all the duplicate rows in an existing table so that you can examine them, you can use a SELECT INTO query that includes a GROUP BY clause and a HAVING clause.

Using UPDATE

The UPDATE Transact-SQL statement changes existing data in a table or view. If you have detected duplicate or dirty data, you need to use UPDATE to replace it with valid data. UPDATE is typically used with the WITH argument, which specifies the temporary named result set or view—also known as common table expression (CTE)—defined within the scope of the UPDATE statement. The CTE result set is derived from a simple query and is referenced by the UPDATE statement.

The TOP argument specifies the number or percent of rows that will be updated. The UPDATE statement also lets you specify server_name, database_name, schema_name, and table_name or view_name as arguments. The view referenced by table_name or view_name must be updatable and must reference exactly one base table in the FROM clause of the view.

Other arguments of the UPDATE statement include: method_name, property_name | field_name, udt_column_name, DEFAULT, expression, column_name, SET, WITH, rowset_function_limited, .WRITE, @variable, <OUTPUT-Clause>, FROM, <table_source>, WHERE, <search_condition>, CURRENT OF, GLOBAL, cursor_name, cursor_variable_name, and OPTION.

The UPDATE statement is logged. However, partial updates to large value data types using the .WRITE clause are logged minimally. UPDATE statements are allowed in the body of user-defined functions only if the table being modified is a table variable.

If an update to a row violates a constraint or rule, or violates the NULL setting for the column, or if the new value is an incompatible data type, the statement is canceled, an error is returned, and no records are updated. When an UPDATE statement encounters an arithmetic error (overflow, divide by zero, or a domain error) during expression evaluation, the update is not performed. The rest of the batch is not executed, and an error message is returned.

If an update to a column or columns participating in a clustered index causes the size of the clustered index and the row to exceed 8,060 bytes, the update fails and an error message is returned. If the UPDATE statement could change more than one row while updating both the clustering key and one or more text, ntext, or image columns, the partial update to these columns is executed as a full replacement of the values.

The setting of the SET ROWCOUNT option is ignored for UPDATE statements against remote tables and local and remote partitioned views. A positioned update using a WHERE CURRENT OF clause updates the single row at the current position of the cursor. This can be more accurate than a searched update that uses a WHERE <search_condition> clause to qualify the rows to be updated. A searched update modifies multiple rows when the search condition does not uniquely identify a single row.

The following example does not use a WHERE clause and increases the value in the ListPrice column by 50 percent for all rows in the Product table:

```
USE AdventureWorks;
GO
UPDATE Production.Product
SET ListPrice = ListPrice * 1.5;
GO
```

The next example uses the WHERE clause to specify which rows to update. Adventure Works Cycles sells its bicycle model Road-250 in two colors: red and black. The company has decided to change the color of red for this model to metallic red. The following statement updates the rows in the Production.Product table for all red Road-250 products:

```
USE AdventureWorks;
GO
UPDATE Production.Product
SET Color = N'Metallic Red'
WHERE Name LIKE N'Road-250%' AND Color = N'Red';
GO
```

MORE INFO UPDATE

For more information about and examples of the UPDATE statement, search for UPDATE (Transact-SQL) in Books Online or access msdn.microsoft.com/library/default.asp?url=/library/en-us/tsqlref/ts_ua-uz_82n9.asp.

Using CHECK Constraints

A CHECK constraint specifies a Boolean search condition that the database engine applies to all values entered in a column. The search condition evaluates to TRUE, FALSE, or unknown. All entries that evaluate to FALSE are rejected.

You can use CHECK constraints to ensure that users enter column data that adheres to a specified format in a column. You could, for example, ensure that all part numbers on a Production table consist of two letters followed by four numbers. This does not ensure the data is accurate, but it does ensure it is in the correct format.

You can create a CHECK constraint as part of a table definition when you create a table. The following example creates the CHECK constraint chk_part_no in the part_no column of the stock_items table. The part_no column is the primary key (PK), although a constraint can be defined for any column. The constraint ensures that users can enter only numbers within a specified range for the key:

```
USE AdventureWorks;
GO
CREATE TABLE stock_items
(
    part_no    int    PRIMARY KEY,
    part_name  char(50),
    part_price money,
    CONSTRAINT part_no CHECK (part_no BETWEEN 1000 and 9999)
);
GO
```

Tables and columns can contain multiple CHECK constraints. If a table already exists, you can use the Transact-SQL ALTER TABLE statement to add, modify, drop, enable, or disable a CHECK constraint. To modify a CHECK constraint, you must first use ALTER TABLE to delete the existing CHECK constraint and then re-create it with a new definition.

MORE INFO Constraints

In addition to using CHECK constraints, you can also define UNIQUE, PRIMARY KEY, and FOREIGN KEY constraints. UNIQUE constraints enforce the uniqueness of the values in a set of columns. PRIMARY KEY constraints identify the column or set of columns that have values that uniquely identify a row in a table. FOREIGN KEY constraints identify and enforce the relationships between tables. For more information, search for “Constraints” in Books Online.

Using CHECKSUM

The CHECKSUM Transact-SQL function returns the checksum value computed over a row of a table or over a list of expressions. CHECKSUM is intended for use in building hash indexes, which improve indexing speed when the column to be indexed is a long character column. The checksum index can also be used for equality searches.

Using CHECKSUM with the star (*) argument specifies that computation is over all the columns of the table. CHECKSUM returns an error if any column is of a noncomparable data type. Noncomparable data types are text, ntext, image, and cursor, and also sql_variant with any one of the preceding types as its base type. You can also use CHECKSUM with an argument that specifies an expression of any type except a non-comparable data type.

CHECKSUM computes a hash value of type int—called the checksum—over its list of arguments. If the arguments to CHECKSUM are columns and an index is built over the computed CHECKSUM value, the result is a hash index. This can be used for equality searches over the columns.

If you apply CHECKSUM over any two lists of expressions, it returns the same value when the corresponding elements of the two lists have the same type and are equal when compared using the equals (=) operator. Null values of a specified type are considered to compare as equal. If one of the values in the expression list changes, the checksum of the list also generally changes. However, a small possibility exists that the checksum will not change. A small possibility also exists that two lists return the same checksum when their contents are not identical. Checksum can give a level of confidence that data has not changed or that two columns are equal. It does not indicate a certainty.

The following examples show how CHECKSUM builds hash indexes and how the SELECT statement makes use of them. The first example builds the hash index by adding a computed checksum column to the table being indexed, and then building an index on the checksum column:

```
SET ARITHABORT ON;
USE AdventureWorks;
GO
ALTER TABLE Production.Product
ADD cs_Pname AS CHECKSUM(Name);
GO
CREATE INDEX Pname_index ON Production.Product (cs_Pname);
GO
```

The second example uses the index in a SELECT query and adds a second search condition to catch cases where checksums match but the values are not the same:

```
USE AdventureWorks;
GO
SELECT *
FROM Production.Product
WHERE CHECKSUM(N'Bearing Ball') = cs_Pname
AND Name = N'Bearing Ball';
GO
```

Alternatively, you can build an index directly on the column indexed. However, if the key values are long, a regular index is not likely to perform as well as a checksum index.

Cleaning Data

Earlier in this lesson, we saw that the Fuzzy Lookup and Fuzzy Grouping transformations and SSIS can be used to clean dirty data caused by errors in data entry or by obtaining data from disparate sources. However, cleaning data is a complex operation, and typically you would use a software package to perform the task. SQL Server 2005 provides the Data Cleaning sample.

The Data Cleaning sample is a package that uses data consisting of a list of names and addresses that represent potential customers. The data contains spelling errors, is missing information, and includes customers already in the database, spurious customers, or multiple instances of the same customer.

The package control flow consists of two tasks. The first is an Execute SQL task that creates the input table, CustomerLeads, and three output tables, named ExistingCustomerLeads, NewCustomerLeads, and DuplicateCustomerLeads. The second is a data

flow task that executes the data flow that performs the cleaning of data extracted from the CustomerLeads table. The data flow identifies unique new, existing, and duplicate customers, and it writes the rows of each customer type to the appropriate output table.

If the samples were installed to the default installation location, the Data Cleaning package is located in the folder C:\Program Files\Microsoft SQL Server\90\Samples\Integration Services\Package Samples\DataCleaning Sample\Data Cleaning\.

Table 9-2 lists the files required to run the sample package.

Table 9-2 Data Cleaning Sample Files

File	Description
DataCleaning.dtsx	The sample package
CreateTables.sql	SQL statements to create tables

You can run the package from the command line by using the *dtexec* utility, or you can run it in BIDS.

To run the package by using *dtexec*, open a Command Prompt window, change the directory to C:\Program Files\Microsoft SQL Server\90\DTS\Binn (the location of *dtexec*), and enter the following command:

```
dtexec /f "C:\Program Files\Microsoft SQL Server\90\Samples\Integration  
Services\Package Samples\Data Cleaning Sample\DataCleaning  
\DataCleaning.dtsx"
```

To run the package in BIDS, open BIDS, point to Open on the File menu and choose Project/Solution. Locate the DataCleaning Sample folder, and then double-click the file named DataCleaning.sln. In Solution Explorer, right-click DataCleaning.dtsx in the SSIS Packages folder, and then choose Execute Package.

MORE INFO Data Cleaning sample

For more information about this sample package, search for "Data Cleaning Package Sample" in Books online or access msdn2.microsoft.com/en-us/library/ms160742.aspx. For more information about how to install samples, search for "Installing Sample Integration Services Packages" in Books Online or access [msdn2.microsoft.com/en-us/library/ms160898\(d=ide\).aspx](http://msdn2.microsoft.com/en-us/library/ms160898(d=ide).aspx).

Discarding Duplicates

You can discard duplicates from the results of a query that uses the Transact-SQL `SELECT` statement by specifying the `DISTINCT` clause. However, this strategy does not represent a good solution for removing duplicate rows from a database table. It operates, typically, on columns, and you would use it, for example, if you wanted to know what product types and how many product types were listed in a table. A table can, however, quite legitimately hold several unique rows—each describing a single product—that have the same product type. Also, the `SELECT` statement displays results. You need to take further action if you want to feed these results back into the database.

As with data cleaning, removing duplicates is a complex process and is typically implemented by using a software package. The process is complicated by the constraint that discarding duplicate rows is usually unwise. Instead, you should create a table to store duplicates so that you can inspect them individually. SQL Server 2005 provides the Remove Duplicates sample to illustrate how duplicates can be removed and stored.

The Remove Duplicates sample demonstrates the implementation of a data flow transformation component with asynchronous outputs. Components with asynchronous outputs receive an input and output *PipelineBuffer* corresponding to the input and output of the object. The input buffers contain rows provided by upstream components. The output buffer is empty and is filled by the component, typically using the rows from the input buffer, during a call to the *ProcessInput* method. After all the rows have been received, they are sorted, and then the distinct rows are sent to one output and the duplicate rows to the other.

MORE INFO *PipelineBuffer* class

The *PipelineBuffer* class provides an in-memory data store containing rows and columns of data. For more information, search for “PipelineBuffer Class” in Books Online or access [msdn2.microsoft.com/en-us/library/microsoft.sqlserver.dts.pipeline.pipelinebuffer\(d=ide\).aspx](http://msdn2.microsoft.com/en-us/library/microsoft.sqlserver.dts.pipeline.pipelinebuffer(d=ide).aspx).

MORE INFO *ProcessInput* script

The *ProcessInput* script component processes the inputs in script components such as transformations and destinations that receive inputs from upstream components. For more information, search for “ScriptComponent.ProcessInput Method” in Books Online or access [msdn2.microsoft.com/en-us/library/microsoft.sqlserver.dts.pipeline.scriptcomponent.processinput\(d=ide\).aspx](http://msdn2.microsoft.com/en-us/library/microsoft.sqlserver.dts.pipeline.scriptcomponent.processinput(d=ide).aspx).

NOTE Using Advanced Editor

The Integration Services Data Flow Programming code samples are intended to demonstrate the core functionality that you need to implement to create a custom data flow component. The samples do not include full support for customization in the Advanced Editor. For example, you cannot use the Advanced Editor to add or remove inputs and outputs or to configure columns.

IMPORTANT Do not use this sample in a production environment.

Microsoft provides the Remove Duplicates sample for educational purposes only. It has not been tested in a production environment. Microsoft does not provide technical support for this sample.

This Remove Duplicates sample requires that you install the .NET Framework software development kit (SDK) 2.0 or later, or Visual Studio 2005. You can obtain the .NET Framework SDK free of charge at the Microsoft Download Center. Access www.microsoft.com/downloads/details.aspx?FamilyID=fe6f2099-b7b4-4f47-a244-c96d69c35dec&displaylang=en.

If you installed the code samples to the default installation location, the C# version of the code sample is located in the following folder:

```
C:\Program Files\Microsoft SQL Server\90\Samples\Integration
Services\Programming Samples\Data Flow\RemoveDuplicates Component
Sample\CS\RemoveDuplicates
```

If the code samples were installed to the default installation location, the Visual Basic version of the code sample is located in the following folder:

```
C:\Program Files\Microsoft SQL Server\90\Samples\Integration
Services\Programming Samples\Data Flow\RemoveDuplicates Component
Sample\VB\RemoveDuplicates
```

To build the sample in Visual Studio 2005, change your directory to the install directory and enter **sn -k keypair.snk** at a .NET Framework or Visual Studio 2005 command prompt. From the File menu, choose Open, choose Project, and then open Remove-Duplicates.sln in the desired solution directory. Press F5 or select Start from the Debug menu to compile and run the project.

To build the sample using the command-line compiler, change the directory to the sample directory and enter **sn -k keypair.snk** at a .NET Framework or Visual Studio 2005 command prompt. Change the directory to the desired solution directory, and type the following command to build either the C# or the Visual Basic version of the sample:

```
for /r %f in (*.sln) do msbuild.exe "%f"
```

The sample is provided in both Visual Basic and C#. To distinguish the assemblies for each version of the sample, the name of the output assembly has CS or VB appended. After you have successfully built the component, add it to a data flow task in BIDS by copying the assembly (*DataSetDestinationCS.dll* or *DataSetDestinationVB.dll*) to the PipelineComponents folder located at %system%\Program Files\Microsoft Sql Server\90\Dts. Open the directory where the global assembly cache (GAC) is located, at %system%\assembly. Select the assembly in the first window, and then drag and drop it into the window containing the GAC. Add the component to the Data Flow Sources section of the Toolbox in BIDS.

To install the component into the global assembly cache (GAC) by using gacutil.exe, open a Command Prompt window and type the following command to run gacutil.exe and install the C# version of the component into the GAC:

```
gacutil.exe -if "c:\Program Files\Microsoft Sql  
Server\90\DTS\PipelineComponents\ RemoveDuplicatesCS.dll"
```

Type the following command to run gacutil.exe and install the Visual Basic version of the component into the GAC:

```
gacutil.exe -if "c:\Program Files\Microsoft Sql  
Server\90\DTS\PipelineComponents\ RemoveDuplicatesVB.dll"
```

Finally, add the component to the Data Flow Sources section of the BIDS Toolbox. After you complete these steps, the component should be visible in the Data Flow Items tab of the Toolbox, and it can be added to the data flow task in SSIS Designer.

When you have added the component to the data flow task and connected it to an upstream component that will provide rows, select the columns used by the component on the Input Columns tab of the Advanced Editor. Only the selected columns are passed to the next component in the data flow. The contents of each column are compared to determine whether a row matches other rows.

MORE INFO Remove Duplicates component sample

For more information, search for "Remove Duplicates Component Sample" in Books Online or access [msdn2.microsoft.com/en-us/library/ms160916\(d=ide\).aspx](https://msdn2.microsoft.com/en-us/library/ms160916(d=ide).aspx).

PRACTICE Using the UNION Operator

In this practice session, you create a table called *dbo.Headlights* by extracting information from the Production.ProductModel table in the AdventureWorks Database. You then extract information from both the Production.ProductModel and the *dbo.Headlights*

table and combine this information into one result set by using the UNION operator. The practice session demonstrates how you can extract information to a database table and how you can combine information from several sources. Do not attempt Practice 2 until you have completed Practice 1.

► **Practice 1: Creating the dbo.Headlights Table**

In this practice, you first delete any existing table called dbo.Headlights. You then create and display the dbo.Headlights table.

1. Log in to your domain at your member server by using either a domain administrator account or an account that has been added to the sysadmin server role. If you are using virtual machine software, log in to your domain and connect to your member server.
2. From the All Programs (or Programs) menu, choose Microsoft SQL Server 2005, and then choose SQL Server Management Studio.
3. In the Connect To Server dialog box, specify Database Engine as the server type. Specify the name of your member server as the server name. Specify Windows Authentication. On the Options tab, specify the AdventureWorks Database and the TCP/IP protocol. Click Connect.
4. Click New Query.
5. In Query Editor, type in the following code:

```
USE AdventureWorks;
GO
IF OBJECT_ID ('dbo.Headlights', 'U') IS NOT NULL
DROP TABLE dbo.Headlights;
GO
```

6. Click Execute. This query drops any existing table called dbo.Headlights. Check that the commands completed successfully as shown in Figure 9-5.
7. In Query Editor, type the following code:

```
USE AdventureWorks;
GO
SELECT ProductModelID, Name
INTO dbo.Headlights
FROM Production.ProductModel
WHERE ProductModelID IN (109, 110);
GO
```

8. Click Execute. This query creates the dbo.Headlights table as shown in Figure 9-6. The table has two rows.

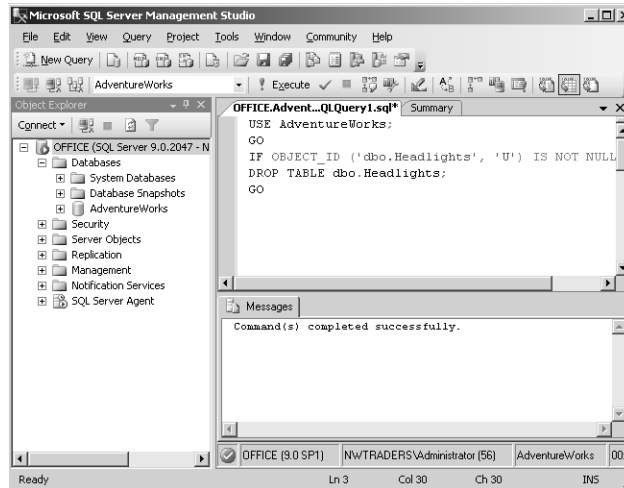


Figure 9-5 Dropping dbo.Headlights.

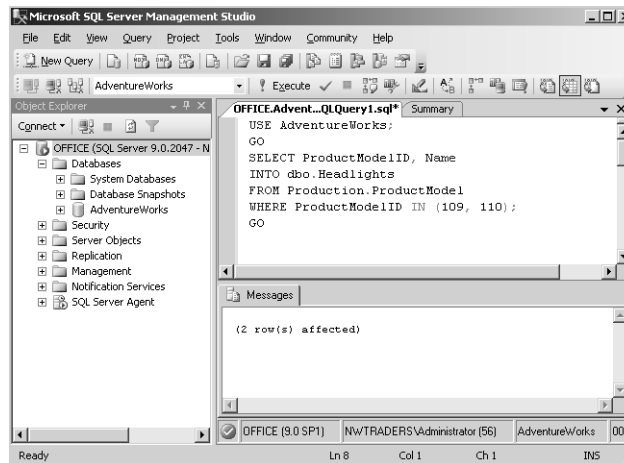


Figure 9-6 Creating dbo.Headlights.

- In Query Editor, type the following code:

```
USE AdventureWorks;  
GO  
SELECT *  
FROM dbo.Headlights;  
GO
```
- Click Execute. This query returns the dbo.Headlights table as shown in Figure 9-7.

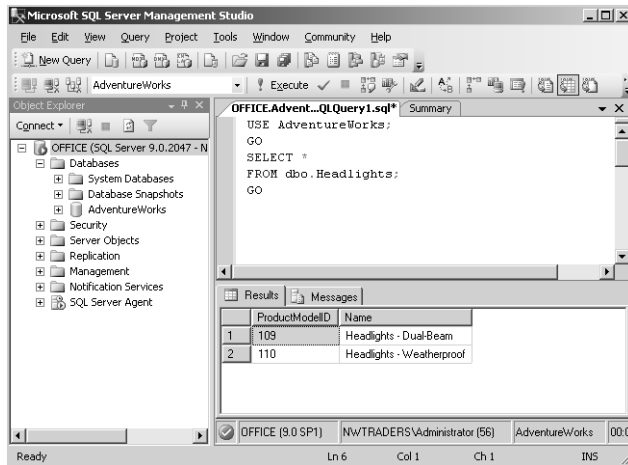


Figure 9-7 The dbo.Headlights table.

► Practice 2: Extracting Information from Two Tables in One Result Set

You should not attempt this Practice until you have successfully completed Practice 1. If you are carrying out the Practice directly after Practice 1, go to step 1 of the Practice. Otherwise, log in, connect SSMS to your member server, and start a new query as described in steps 1 through 4 of Practice 1.

1. In Query Editor, type the following code:

```

USE AdventureWorks;
GO
SELECT ProductModelID, Name
FROM Production.ProductModel
WHERE ProductModelID NOT IN (109, 110)
UNION
SELECT ProductModelID, Name
FROM dbo.Headlights
ORDER BY Name;
GO

```

2. Click Execute. This query returns rows from both the Production.ProductModel and dbo.Headlights tables as shown in Figure 9-8. Note that the rows from the dbo.Headlights table are not displayed separately or out of order. The rows are ordered by the contents of the Names column.

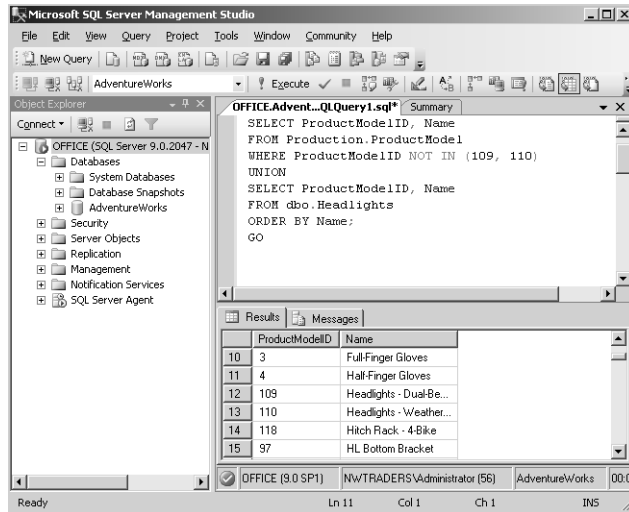


Figure 9-8 Rows from both tables.

Lesson Summary

- You need to consult with business analysts, managers, and users to gather business and regulatory requirements, to determine the business requirements for data, and to generate a definition of data quality.
- Techniques such as duplicate removal and field value standardization, and the use of data validation and fuzzy logic, can help address the problems associated with badly entered data.
- Data mining and analysis determines what you are actually doing with the data. These techniques can provide models and predictions that enable managers to make business decisions.
- Microsoft provides a wide variety of tools to enable application developers to build connected systems and applications that meet modern requirements for the quality, cohesiveness, availability, and security of data.
- You can create queries to inspect data by using left and right outer joins, full outer joins, and unions; and by using the HAVING clause. You can use binary checksum to check data replicated or copied to another location. You can clean data by discarding duplicates and by using the UPDATE statement.

Lesson Review

You can use the following questions to test your knowledge of the information in Lesson 1, “Enforcing Data Quality According to Business Requirements.” The questions are also available on the companion CD if you prefer to review them in electronic form.

NOTE Answers

Answers to these questions and explanations of why each answer choice is correct or incorrect are located in the “Answers” section at the end of the book.

1. Your organization uses a SQL Server 2005 database that includes a table named Products. The table includes a column named ProductCode that contains string data. Written company policy specifies the format for data in the ProductCode column. Users can use ad hoc queries to enter data into the ProductCode column. You need to ensure that the data in the ProductCode column is properly formatted. What should you do?
 - A. Use a UNIQUE constraint.
 - B. Use the UPDATE statement.
 - C. Use the HAVING clause.
 - D. Use a CHECK constraint
2. Which one of the following Business Intelligence (BI) tools should you use for extracting, transforming, and loading data?
 - A. SSMS
 - B. BIDS
 - C. SSIS
 - D. SSRS
3. You are the senior DBA at Litware, Inc. You use BIDS to create an SSIS package in your development environment. You then use the SSIS package to import data into your development environment from one of Litware’s trading partners. You need to deploy the SSIS package to your production environment. Your production databases use table names that are different from those used in your development environment. What should you do?

- A. Create a SQL Server Integration Services (SSIS) package configuration. Build a deployment utility. Copy the deployment folder for your SSIS project to your production server. Execute the manifest file.
 - B. Use the UNION statement to combine the information in the production and development databases.
 - C. Back up the msdb database, and restore it to the production server. Rename the appropriate tables inside the msdb database.
 - D. Use the ALTER TABLE Transact-SQL statement in a script file to change the table names in your development databases so that they are the same as those in your production database.
4. Which of the following tasks can the Fuzzy Lookup transformation perform? (Choose all that apply.)
- A. Identify rows of data that are likely to be duplicates.
 - B. Tokenize the data.
 - C. Specify the maximum number of matches to return per input row.
 - D. Use similarity thresholds to derive similarity and confidence scores.
5. An SSIS package on your SQL Server 2005 server typically takes five minutes to run. You discover it is currently taking almost half an hour, and you suspect that memory is too low. You need to identify the problem. What should you do?
- A. Check the SSIS package manifest files to determine whether they specify any in-built delays.
 - B. Use the deployment utility to redeploy the SSIS package.
 - C. Use System Monitor to monitor all the counters associated with the SQL Server:SSIS Pipeline performance object.
 - D. Use Profiler to create traces that specify event classes in the Security Audit Event Category.

Lesson 2: Optimizing a Database Change Control Strategy to Meet Business Requirements

Your function as a DBA is not only technical; it also involves managerial responsibilities. You must be aware of the need to control changes implemented on test or pre-production networks and migrated to production servers. You need to implement and enforce change control procedures for the various stages of database operations, and work with database and application developers to ensure that all such changes are documented and a paper trail exists.

This lesson discusses the managerial aspect of database change control. It also discusses the technical tools that you can use to prevent specified changes or to determine what changes have been made and who made them. You need to be able to identify the database objects related to a deployment, control schema changes, and use the most efficient method to migrate changes to production networks. The lesson covers all these topics.

After this lesson, you will be able to:

- Set up systems that help you verify that change control procedures are being followed.
- Use DDL Triggers to limit the changes that can be made to the structure of a database.
- Set security permissions to enable other DBAs to perform required tasks and to check that they are following the correct procedures.
- Use stored procedures to identify database objects related to a particular deployment.
- Administer schema changes.
- Use migration scripts.

Estimated lesson time: 45 minutes

Verifying that Database Change Control Procedures Are Being Followed

As a DBA, one of your many core tasks is change control management. You need to implement change control procedures for the various stages of database operations, from the development stage right through to supporting the production environment. You can summarize the stages as follows:

- **Development** Change control needs to take account of application development in addition to database structure. The database structure and associated

applications go through a large number of changes, some of which do not go to the test stage and some of which go all the way through to production.

- **Test** Change control involves ongoing user acceptance testing of new applications and structures from the development stage, and feedback from the production stage that requires further testing of database implementations. At this stage, database security privileges reflect those in force (or that will be in force) in the production environment.
- **Production support** Provides a mirror of the production environment at a given point in time for the testing of fixes or debugging of critical problems. At this stage, change control involves batches of changes that are either discarded or implemented on production servers.
- **Pre-production** Provides a mirror of the production environment that developers use when compiling code and for the final pre-testing of production changes. Change control at this stage should feed back into the test or production support stages. The database and application designers (working as a team with DBAs, system administrators, and end users) need to test and debug changes before implementing them on production servers.
- **Production** Except in a genuine emergency (a hotfix), changes should not be made directly to the production environment. Change control requests should feed back to the test or production support stages. Change control requests feed forward to the production stage only when changes have been tested and debugged.

A production database of any significant size is seldom a static entity. Feedback from users, new requirements, new sources of data, new developments, and incident requests all result in changes that feed back and forward through the stages. Change control management requires that all change requests and all changes be documented and have signed approval, and that a paper trail exists so that developers and DBAs can track the changes. Users need to be confident that incident (bug) reports and service requests receive responses within the time that service level agreements (SLAs) specify.

Visual SupportSafe

Source control software is a key item that underpins any development project. A variety of products exists, but a large number of sites use Microsoft Visual SourceSafe (VSS), which is integrated with SQL Server 2005. A well-managed and secure VSS database is critical to ongoing source management.

If you choose VSS as your source control software, you need to understand how to use labeling and pinning, along with the process of sharing files and repinning. Many users make complete separate copies for each working folder or (worse still) create a single working folder for development, test, and production source code. If users consider that pinning, labeling, branching, and so on are too complex, you can create either three separate VSS databases containing development, test, and production source code, or three project folders. Whatever method you choose, users need a disciplined approach to source control management.

Finally, ensure you are using the latest version of the software and apply the latest service packs.

MORE INFO Visual SourceSafe

For more information about VSS, access msdn.microsoft.com/vstudio/previous/ssafe/productinfo/default.aspx.

Using DDL Triggers

DDL Triggers fire stored procedures in response to data definition language (DDL) statements. They can be used to perform administrative tasks in the database such as auditing and regulating operations that create, alter, and delete databases. Unlike data manipulation language (DML) triggers, DDL triggers do not fire in response to UPDATE, INSERT, or DELETE statements on a table or view. Instead, they fire in response to DDL statements—typically, statements that start with CREATE, ALTER, and DROP.

You can use DDL triggers when you want to do the following:

- Prevent certain changes to your database schema.
- Automatically make changes in the database in response to a change in your database schema.
- Record changes or events in the database schema.

The practice session in this lesson illustrates how a DDL trigger can be used to prevent any table in the AdventureWorks database from being modified or dropped.

To create a DDL trigger, you need to specify the DDL trigger scope and determine which Transact-SQL statement, or group of statements, fires the trigger. DDL triggers fire in response to a Transact-SQL event processed in the current database or on the current server. The scope of the trigger depends on the event. For example, a DDL

trigger that you create to fire in response to a CREATE TABLE event does so whenever a CREATE TABLE event occurs in the database. A DDL trigger that you create to fire in response to a CREATE LOGIN event does so whenever a CREATE LOGIN event occurs in the server. In the first instance, you use the ON DATABASE scope; in the second instance, you use the ON ALL SERVER scope.

Database-scoped DDL triggers are stored as objects in the database in which you create them. You can create DDL triggers in the master database, and they behave in a similar way to those created in user-designed databases. You can obtain information about DDL triggers by accessing the *sys.triggers* catalog view from within the database context in which they are created, or by specifying the database name as an identifier (for example, *master.sys.triggers*).

Server-scoped DDL triggers are stored as objects in the master database. You can, however, obtain information about server-scoped DDL triggers by accessing the *sys.server_triggers* catalog view in any database context.

Server-scoped DDL triggers appear in SSMS Object Explorer in the Triggers folder. This folder is located under the Server Objects folder. Database-scoped DDL triggers appear in the Database Triggers folder. This folder is located under the Programmability folder of the corresponding database.

NOTE Temporary tables and stored procedures

DDL triggers do not fire in response to events that affect local or global temporary tables and stored procedures.

You can create DDL triggers to fire in response to one or more DDL statements or a predefined group of DDL statements—for example, you can design a trigger that fires in response to any DROP TABLE or ALTER TABLE event in a specified database. However, you cannot use all DDL events in DDL triggers. For example, you cannot use a CREATE DATABASE event in a DDL trigger. You should instead use event notifications for these events.

MORE INFO Events for use with DDL triggers

For more information about which Transact-SQL statements you can use with DDL Triggers, and the scope at which they fire, access “DDL Events for Use with DDL Triggers” at [msdn2.microsoft.com/en-us/library/ms189871\(d=ide\).aspx](https://msdn2.microsoft.com/en-us/library/ms189871(d=ide).aspx).

A DDL Trigger can fire after execution of any Transact-SQL event that belongs to a predefined grouping of similar events. For example, if you want a DDL trigger to fire after any CREATE TABLE, ALTER TABLE, or DROP TABLE statement is run, you can specify FOR DDL_TABLE_EVENTS in the CREATE TRIGGER statement. After you create the trigger, you can determine the events that the event group covers by accessing the *sys.trigger_events* catalog view.

MORE INFO Event groups for use with DDL triggers

For more information about the predefined groups of DDL statements that are available for DDL triggers, the particular statements they cover, and the scopes at which these event groups can be programmed, access "Event Groups for Use with DDL Triggers" at [msdn2.microsoft.com/en-us/library/ms191441\(d=ide\).aspx](http://msdn2.microsoft.com/en-us/library/ms191441(d=ide).aspx).

IMPORTANT Returning result sets

The ability to return result sets from triggers will be removed in a future version of SQL Server. Triggers that return result sets might cause unexpected behavior in applications that are not designed to work with them. Avoid returning result sets from triggers in new development work, and plan to modify applications that currently do this. To prevent triggers from returning result sets in SQL Server 2005, use the *sp_configure* system stored procedure to set the Disallow Results From Triggers option to 1. The default setting of this option will be 1 in a future version of SQL Server.

If you need to modify the definition of a DDL trigger, you can either drop and re-create the trigger or redefine the existing trigger. If you change the name of an object referenced by a DDL trigger, you need to modify the trigger so that its text reflects the new name. Therefore, before renaming an object, you should display the dependencies of that object to determine whether any triggers are affected by the proposed change. You can use the Transact-SQL stored procedure *sp_depends* to view database object dependencies.

You can also rename any DDL trigger that you own. The database owner (dbo) can, however, change the name of any user trigger in the current database. You can use the Transact-SQL statement ALTER TRIGGER to modify a trigger, and the Transact-SQL stored procedure *sp_rename* to rename it. Renaming a DDL trigger does not change the name of the trigger in the text of the trigger definition. To change the name of the trigger in the definition, you need to modify the trigger.

When you no longer need a DDL trigger, you can disable or delete (drop) it. Disabling a DDL trigger does not drop it. The trigger still exists as an object in the current data-

base. However, the trigger does not fire when any Transact-SQL statements on which it was programmed are run. DDL triggers that are disabled can be re-enabled. Enabling a DDL trigger causes it to fire in the same way it did when it was originally created.

DDL triggers are enabled by default when they are created. When a DDL trigger is deleted, it is dropped from the current database. Any objects or data upon which the DDL trigger is scoped are not affected. You can use the Transact-SQL statements `DISABLE TRIGGER`, `ENABLE TRIGGER`, and `DROP TRIGGER` to respectively disable, enable, and drop a trigger.

Setting Security Permissions

In SQL Server 2005, you can use role-based permissions to restrict access to information or the ability to make changes in your databases. You can control changes more easily if only a limited number of people are permitted to make them, and you can further control the situation by limiting the changes they can make and the databases they can alter. A role is used to control access to data within a given database (database roles) or to delegate certain administrative functions on the server (server roles). Typically, you can control access to your database by using the following types of access control:

- Mandatory Access Control (MAC)
- Discretionary Access Control (DAC)
- Role-Based Access Control (RBAC)

You use MAC in very secure environments (for example, military establishments) where information is classified at a certain level. If you are not at the appropriate level, you do not have access. You use DAC if you want to assign permissions on a per-user basis. SQL Server 2005 also offers RBAC, which can simplify your security planning.

If, for example, you want to give a particular user (possibly a member of your DBA team) the ability to issue a `SELECT` statement against every table and view within a given database, you can add the user's account to the `db_datareader` fixed database role. If you want to give a user the ability to modify data in any table or view within the database, you can add the user's account to the `db_datawriter` fixed database role. If you want to give a user the ability create, alter, drop, and restore any database on a SQL Server 2005 server instance, you can add the user's account to the `dbcreator` fixed server role.

MORE INFO Fixed database and server roles

For more information about fixed database roles and fixed server roles, and the granular SQL Server 2005 permissions to which they map, refer to the sections “Verifying SQL User Accounts Assigned to Database Roles” and “Verifying Permissions on SQL Server Roles and Accounts” in Chapter 11, “Security Strategies.”

You can also control access to databases and tables through the use of views and stored procedures. However, RBAC provides a quick and convenient method of defining which users have permission to perform specific necessary tasks and exactly what tasks they can perform. You have the option of creating a user-defined role that grants a set of permissions that you specify, or of granting permissions to users manually. For example, you can use the CREATE ROLE Transact-SQL statement to create a new database role or the GRANT Transact-SQL statement to grant permissions to a database principal or a database role.

You can audit the use of SQL Server 2005 permissions to help you to verify that users with these permissions are observing database change control procedures. Microsoft SQL Server Profiler can create traces that specify event classes in the Security Audit Event Category. For example, the Audit Database Management event class occurs when a database is created, altered, or dropped. A trace of this event class provides you with the name of the database, the login name of the user, the event subclass (create, alter, drop, and so on), the Windows user name and domain of the user, and the name of the application (if any) that was run against the database to implement the change.

You can specify the Audit Database Object Management event class. This event class occurs when a CREATE, ALTER, or DROP statement is executed on a database object. You can also obtain useful information by specifying the Audit Database Principal Impersonation, which occurs when an impersonation occurs within the database scope—for example, *execute as user* or *setuser*—and the Audit Database Principal Management event class, which occurs when principals—for example, users—are created, altered, or dropped from a database.

MORE INFO Security Audit event category

For more information about the event classes in the Security Audit event category, search for “Security Audit Event Category (SQL Server Profiler)” in Books Online or access msdn2.microsoft.com/en-us/library/ms346021.aspx.

Identifying All Database Objects Related to a Particular Deployment

If you are optimizing a database change control strategy to meet business requirements, you need to identify the database objects related to the particular deployment that you want to optimize. Database objects include servers, databases, tables, views, columns, indexes, triggers, procedures, constraints, and rules. You need to ensure that no ambiguity exists when identifying database objects. Ambiguity can occur when multiple objects have the same name in the system. For example, two tables might have different columns with the same name, or two users might own different tables with the same name.

To resolve ambiguity, you might need to qualify objects. For example, *column* can be qualified as *table.column*, further identified as *database.owner.table.column*, and finally fully qualified as *server.database.owner.table.column*.

Using fully qualified database objects all the time can become cumbersome. You could, therefore, use an alias in the FROM clause for aliased tables or views. The following example uses the alias *Cus* for *MyDatabase.MyOwner.tblCustomer*:

```
SELECT tblCustomer.Name
FROM MyDatabase.MyOwner.tblCustomer AS Cus
     INNER JOIN MyDatabase.MyOwner.tblOrders AS Ord
     ON Cus.Name = Ord.Name
WHERE Ord.Name = 'KimAkerOrder'
```

During the process of gathering information prior to analyzing and possibly altering database deployment, you need to identify the key objects or entities that the database manages. The object can be a tangible thing, such as a person or a product, or it can be a more intangible item, such as a business transaction, a department in a company, or a payroll period. In general, when you have identified a few primary objects, the related items become visible. Each distinct item in your database should have a corresponding table.

For example, the primary object in the AdventureWorks database is a bicycle. The objects related to bicycle within this company's business are the employees who manufacture the bicycle, the vendors that sell components used to manufacture the bicycle, the customers who buy bicycles, and the sales transactions that occur. Each of these objects has a corresponding table in the database.

When you identify the objects in a system, you should record them in a way that provides a visual representation of that system. You can use this database model as a reference during database implementation or alteration. For this purpose, database

developers use tools that range in technical complexity from pencils and scratch paper to word processing and spreadsheet programs, and to software programs created specifically for the job of data modeling for database designs. Whatever tool you decide to use, you need to keep it up to date.

After you identify the primary objects in a database, you need to identify the types of information that must be stored for each object. These are the columns in the table of the object. You can categorize the columns in a database table as follows:

- **Raw data columns** Use this type to store tangible pieces of information, such as names, determined by a source external to the database.
- **Categorical columns** Use this column type to classify or group the data and store a limited selection of data—for example, true/false or male/female.
- **Identifier columns** Use this type to provide a mechanism to identify each item stored in the table. Identifier columns typically have an ID or number in their name. The identifier column is the primary component that users and internal database processing use for gaining access to a row of data in the table. Sometimes the object has a tangible form of ID used in the table—for example, a Social Security number—but in most situations you can define the table so that a reliable, artificial ID can be created for the row.
- **Relational or referential columns** Use this column type to establish a link between information in one table and related information in another table. For example, a table that tracks sales transactions typically has a link to the customers table so that customer information can be associated with the sales transaction.

Using Stored Procedures

You can identify objects related to a deployment by running ad hoc queries or scripts against database tables that contain identifier or relational columns. However, if the database is large or if the list of objects changes frequently, you can use stored procedures to save time and effort.

SQL Server 2005 stored procedures, which you run by executing the Transact-SQL EXECUTE statement, accept input parameters and return multiple values in the form of output parameters to the calling procedure or batch. They contain programming statements that perform operations in the database, including calling other procedures, and they return a status value to a calling procedure or batch to indicate success or failure (and the reason for failure). Unlike functions, stored procedures do not return values in place of their names and they cannot be used directly in an expression.

You can create stored procedures by using the CREATE PROCEDURE Transact-SQL statement. To create procedures, you must have CREATE PROCEDURE permission in the database and ALTER permission on the schema in which the procedure is being created. Stored procedures are schema-scoped objects, and their names must follow the rules for identifiers. You can create a stored procedure only in the current database.

When creating a stored procedure, you should specify the following:

- Input and output parameters to the calling procedure or batch. A stored procedure can take input parameters, return tabular or scalar results and messages to the client, invoke data definition language (DDL) and data manipulation language (DML) statements, and return output parameters. For example, a stored procedure can use an extended markup language (XML) formatted file as an input argument.
- Programming statements that perform operations in the database, including calling other procedures.
- The status value returned to the calling procedure or batch to indicate success or failure (and the reason for failure).
- Any error handling statements needed to catch and handle potential errors.

MORE INFO Error-handling functions

SQL Server 2005 introduces new error-handling functions, such as ERROR_LINE and ERROR_PROCEDURE, that can be specified in the stored procedure. For more information, search for "Using TRY...CATCH in Transact-SQL" in Books Online or access msdn2.microsoft.com/en-us/library/ms179296.aspx.

You can use the Transact-SQL ALTER PROCEDURE and DROP PROCEDURE statements to respectively modify and delete a stored procedure.

Temporary Stored Procedures

You can create private and global temporary stored procedures by adding the hash (#) and double hash (##) prefixes to the procedure name. The single hash denotes a local temporary stored procedure; the double hash denotes a global temporary stored procedure. You can use temporary stored procedures when connecting to legacy versions of SQL Server that do not support the reuse of execution plans for Transact-SQL statements or batches. Applications connecting

to SQL Server 2000 and later should use the *sp_executesql* system stored procedure instead of temporary stored procedures.

Only the connection that created a local temporary procedure can execute it, and the procedure automatically deletes when the connection closes. Any connection can execute a global temporary stored procedure. A global temporary stored procedure exists until the connection used by the user who created the procedure closes and any currently executing versions of the procedure by any other connections complete.

Common language runtime (CLR) stored procedures cannot be created as temporary stored procedures.

You can create Transact-SQL or CLR stored procedures. A Transact-SQL stored procedure is a saved collection of Transact-SQL statements that can take and return user-supplied parameters. For example, a stored procedure might contain the statements needed to insert a new row into one or more tables based on information supplied by a client application. A CLR stored procedure is a reference to a Microsoft .NET Framework CLR method that can take and return user-supplied parameters.

IMPORTANT Extended stored procedures

This feature will be removed in a future version of SQL Server. Microsoft recommends that you avoid using this feature in new development work and plan to modify applications that currently use the feature. You can use CLR stored procedures instead.

If you frequently access specific information in a database table, you can create a stored procedure to perform this function, as shown in the following example, which creates and then executes the *usp_GetAllEmployees* stored procedure:

```
USE AdventureWorks;
GO
CREATE PROCEDURE HumanResources.usp_GetAllEmployees
AS
SELECT LastName, FirstName, JobTitle, Department
FROM HumanResources.vEmployeeDepartment;
GO
EXECUTE HumanResources.usp_GetAllEmployees;
GO
```

You can use stored procedures to manipulate database objects. A database object represents the properties of a single instance of SQL Server. The database object is a major component of the SQL Distributed Management Objects (SQL-DMO) object

tree and contains collections that define the tables, stored procedures, data types, and users in a database. Methods of the database object enable you to perform the following database management functions by using scripts or stored procedures:

- Create a SQL Server 2005 database.
- Add database roles, rules, stored procedures, tables, user-defined data types, users, and views to an existing SQL Server 2005 database.
- Remove or drop database objects (tables, views, and so on) from an existing SQL Server 2005 database.
- Modify the disk resource that the database uses for storage.
- Back up or restore an existing SQL Server 2005 database or its transaction log.
- Control SQL Server 2005 database security by adding users and granting, denying, or revoking access rights to the database.
- Check SQL Server 2005 database integrity.
- Check current usage in the database. For example, you can check the status of locks applied against database resources.

MORE INFO Database object

For more information, search for “Database Object” in Books Online.

Implementing Schema Changes

Database schemas contain tables, columns, data types, functions, stored procedures, views, and so on. Schemas change to accommodate the varying needs of the organization and require change control. If an organization targets new markets, alters its mission statement, or modifies its business requirements, its databases and database schemas need to change. Changes in applications can also require schema changes. These changes require careful management. All schema changes need to be performed by the DBA team, and the policy for schema change requests is different based on whether the schema relates to a development database or a production database.

For a development database, the DBA team typically holds meetings with the developers to discuss requirements and application specifications. The developers also need to inform the DBA team about future implications, expectations, and requirements. When all parties agree the schema design, the DBA team creates documentation (possibly an e-mail) that describes the proposed schema changes. The developers then generate official documentation requesting the schema changes.

When the DBA team implements the schema changes, it informs the developers. The developers then check and verify the changes and inform the DBA team whether the changes are good and do not require a review. Until and unless the DBA team receives this verification, schema changes are considered to be temporary and are not propagated to the production database.

IMPORTANT A paper trail must exist.

All communications between the developers and the DBA team must be implemented either by e-mail or by signed forms. A paper trail, either electronic or physical, must exist at all stages.

Except in the case of an extreme emergency when a hot fix is required, schema changes will not be made to the production database unless they have first been tested on the development database. When developers want schema changes propagated to the production database, they need to send a communication by a signed form or by e-mail requesting the change. The communication should refer to changes to the development database.

The DBA team responds with details of the permanent changes—the changes for which the developers have returned a signed form—available on the development database. The developers respond by confirming that the listed changes are the ones required. The DBA team responds with a schedule for propagating the changes, and it follows this with a second communication when the changes are made.

The line managers of the development and DBA teams are responsible for enforcing this policy and ensuring a paper trail exists that documents every stage in the operation.

Using Migration Scripts

Most organizations have a range of development stages. They might, for example, have a development stage, test stage, pre-production stage, and production stage. When database changes are implemented and tested at the test or pre-production stages, and after the database design team signs off the changes and requests their implementation in the production environment, you (or your DBA team) need to migrate the database changes and, possibly, the new database structure. You can accomplish this migration by using the following methods:

- Delete the target database, and replace it with the new one. The advantage of this approach is that it is simple. The major drawback is that any data in the replaced database will be lost.

- Use SSMS to migrate changes from the old database to the new one. In this scenario, you manually add and modify database objects using the SSMS user interface. This process is tedious, and it is difficult to document and repeat.
- Create a migration script to convert one database's structure to match that of another. This process involves creating batches of SQL statements that contain commands to make the database schema of the target database the same as that of the source database. This is typically the preferred option.

Creating a migration script allows you to push changes from a single source database to multiple target databases without data loss. You can document the changes between database versions and produce an audit trail that developers or other DBAs can track and reproduce.

Migration scripts can be created either as each database object is modified throughout the development life cycle or prior to the actual migration—after database development is complete. The first option relies on developers and DBAs remembering to create accurate, debugged batches of SQL statements whenever they change a database object. Then somebody (probably you) needs to collate all the changes for the ultimate migration. In practice, creating the scripts at the end of development is usually the easiest and best method.

You also need to make sure the scripts are complete. If you do not push all the changes from one database to the other, the resulting errors will be difficult to find and fix. If the migration scripts are inaccurate, an application might work on the test network but not on the production network.

If you are administering large databases and complex applications run against them, tracking down issues caused by (for example) mismatched database schemas can take considerable time. Such problems are often not detected until late in the software development life cycle, and they are usually difficult to diagnose and fix.

Another problem can arise from the complexity of detecting differences in databases that might contain thousands—or even tens of thousands—of tables, stored procedures, views, and other database objects. This task is almost impossible to do manually. You can use SSMS, but it is not designed specifically for this purpose and you might prefer to use a third-party tool—for example, Red Gate Software's SQL Compare, Embarcadero Change Manager, or AllFusion ERwin Data Modeler.

Real World

Ian McLean

You can use a language such as Visual Basic or Visual C++ in combination with Microsoft Excel spreadsheets to create migration scripts. If you have specific requirements and enjoy writing code, I would not actively discourage you. Although DBAs do not typically write complex and lengthy programs—this is the domain of the developer or applications programmer—and the 70-444 examination is unlikely to test complex coding skills, I always encourage DBAs to write scripts, procedures, and reasonably complex ad hoc queries. If you find yourself repeating the same task on a regular basis, try to automate it. If you're known as the DBA who can knock out a bit of code when required, this does no harm at all to your promotion prospects.

Nevertheless, limits do exist, and I draw the line at writing scripts that migrate large numbers of objects, often following complex rules, especially when a decent automated third-party tool can do the job for you. If your changes do not work properly in the target database, is the fault in your data structures or your home-grown migration script? My advice, with which you are, of course, at liberty to disagree, is to use a reputable automated tool for this purpose.

A good automated tool provides all of the following functions:

- **Gives visual indications of differences in database objects** You need to be able to see, at a glance, how your databases differ before you migrate changes.
- **Creates migration scripts for all database objects** You often need to migrate more than tables, stored procedures, and views. If the permissions on objects differ or tables have different triggers, application errors could result.
- **Creates scripts in the correct order, taking dependencies and foreign keys into account** Database objects need to be dropped, altered, and created in the correct order or database updates will fail. If you alter the data type of a column that is part of an index, you need to drop and re-create the index.
- **Enables you to define the criteria for defining database objects as different** For example, you might consider white space in stored procedure definitions to be insignificant, or you might choose to ignore the collation settings of character columns. Fill factors on indexes, the names of constraints, or the order of columns in tables might not be important to you. The tool should allow you to easily define all these parameters.

MORE INFO Automated migration tools

For more information about Red Gate Software's SQL Compare, Embarcadero Change Manager, and AllFusion ERwin Data Modeler, access www.red-gate.com, www.embarcadero.com/products/changemanager/index.html, and www3.ca.com/solutions/Product.aspx?ID=260, respectively.

PRACTICE Using DDL Triggers

In this practice session, you create a DDL trigger that prevents users from dropping or altering any tables in the AdventureWorks database. You test the trigger, and then remove it. If you create the trigger, make sure that you remove it; otherwise, practices in other chapters might not work. In the second practice, you access the DDL trigger example in the AdventureWorks sample database. You should examine the code. The syntax is correct, but the code does not run by default because an object named “ddl-DatabaseTriggerLog” already exists in the database. If you want to extend the practice, debug this problem.

► Practice 1: Creating, Testing, and Removing a DDL Trigger

In this practice, you create, test, and remove a DDL trigger that prevents users from dropping or altering tables in the AdventureWorks database.

1. Log on to your domain at your member server by using either a domain administrator account or an account that has been added to the sysadmin server role. If you are using virtual machine software, log in to your domain and connect to your member server.
2. From the All Programs (or Programs) menu, choose Microsoft SQL Server 2005, and then choose SQL Server Management Studio.
3. In the Connect To Server dialog box, specify Database Engine as the server type. Specify the name of your member server as the server name. Specify Windows Authentication. On the Options tab, specify the AdventureWorks Database and the TCP/IP protocol. Click Connect.
4. Click New Query.
5. In Query Editor, type the following:

```
USE AdventureWorks;
GO
CREATE TRIGGER tableprotect
ON DATABASE
FOR DROP_TABLE, ALTER_TABLE
AS
PRINT 'You must disable the tableprotect trigger to drop or alter tables!'
ROLLBACK;
```

- Click Execute. Check that the commands completed successfully, as shown in Figure 9-9.

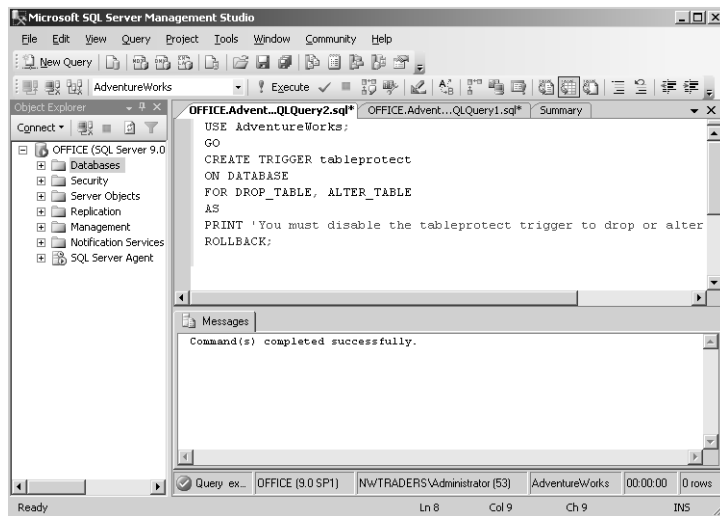


Figure 9-9 Creating the trigger.

- Test the trigger by creating a table in the AdventureWorks database and then attempting to alter it. To do this, type the following in Query Editor:

```
USE AdventureWorks;
GO
CREATE TABLE MyTable (Number INT, Name VARCHAR(20) NULL);
GO
ALTER TABLE MyTable DROP COLUMN Name;
GO
```

- Click Execute. Check that the table was created but could not be altered, as shown in Figure 9-10.
- Drop the trigger by typing the following in Query Editor:


```
Use AdventureWorks;
GO
DROP TRIGGER tableprotect
ON DATABASE;
GO
```
- Click Execute. Check that the commands completed successfully as shown in Figure 9-11.

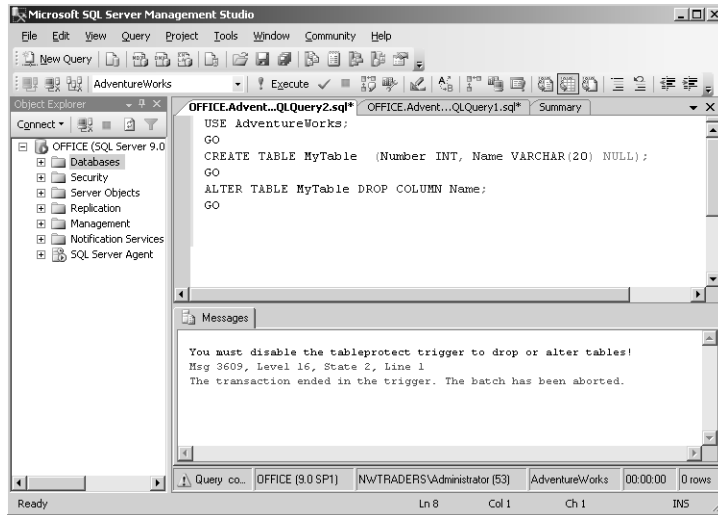


Figure 9-10 A trigger prevents a table alteration.

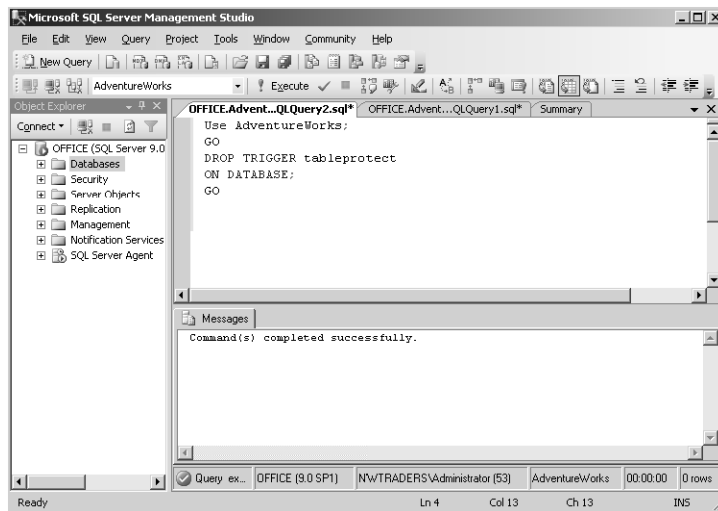


Figure 9-11 Dropping the trigger.

11. Check that the trigger has been dropped by dropping the table. To do this, type the following in Query Editor:

```
Use AdventureWorks;
GO
DROP TABLE MyTable
GO
```

12. Click Execute. Check that the commands completed successfully as shown in Figure 9-12.

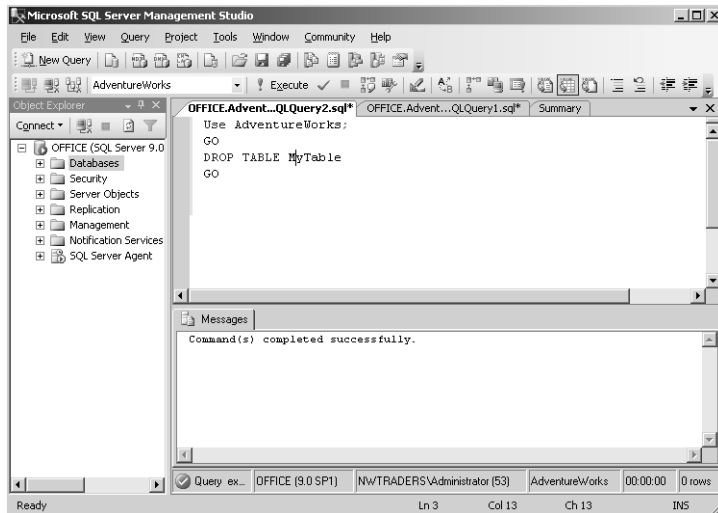


Figure 9-12 Dropping the table.

► Practice 2: Obtaining a DDL Trigger Example

In this practice, you load a DDL trigger example that Microsoft provides for the AdventureWorks sample database. The syntax for this example is correct, but it might not run because the trigger is already present in the database. If you want to take this practice further, debug this problem and create the trigger. If you carry out this practice directly after Practice 1, the first three steps are not required.

1. Log in to your domain at your member server by using either a domain administrator account or an account that has been added to the sysadmin server role. If you are using virtual machine software, log in to your domain and connect to your member server.
2. From the All Programs (or Programs) menu, choose Microsoft SQL Server 2005, and then choose SQL Server Management Studio.
3. In the Connect To Server dialog box, specify Database Engine as the server type. Specify the name of your member server as the server name. Specify Windows Authentication. On the Options tab, specify the AdventureWorks Database and the TCP/IP protocol. Click Connect.
4. In Object Explorer, expand Databases, expand AdventureWorks, expand the Programmability folder, and expand Database Triggers as shown in Figure 9-13.
5. Right-click `ddlDatabaseTriggerLog` and choose Script Database Trigger As. Figure 9-14 shows this selection.

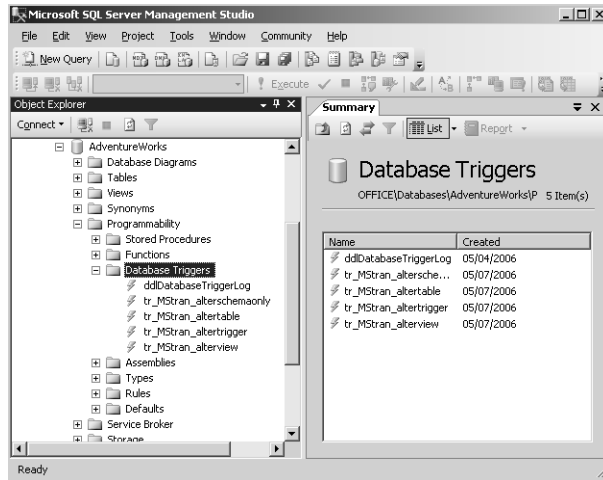


Figure 9-13 Expanding Database Triggers.

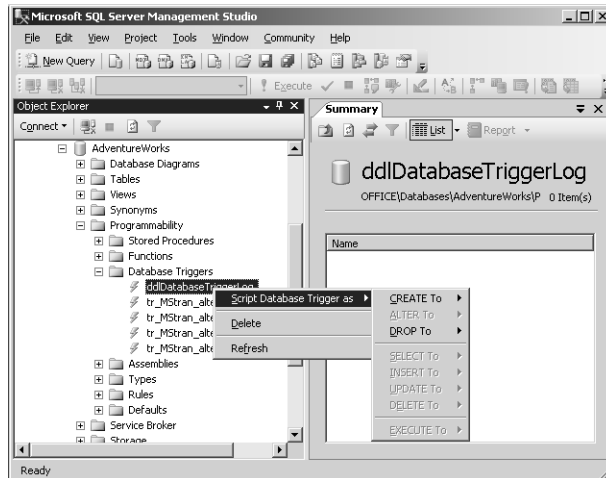


Figure 9-14 Selecting Script Database Trigger As.

- Choose CREATE To and then choose New Query Editor Window. The commands to create the trigger appear in Query Editor as shown in Figure 9-15.
- Read the query. If you come across any commands you have not seen before, look them up in Books Online.
- Parse the commands (blue tick). The syntax should be correct as shown in Figure 9-16.

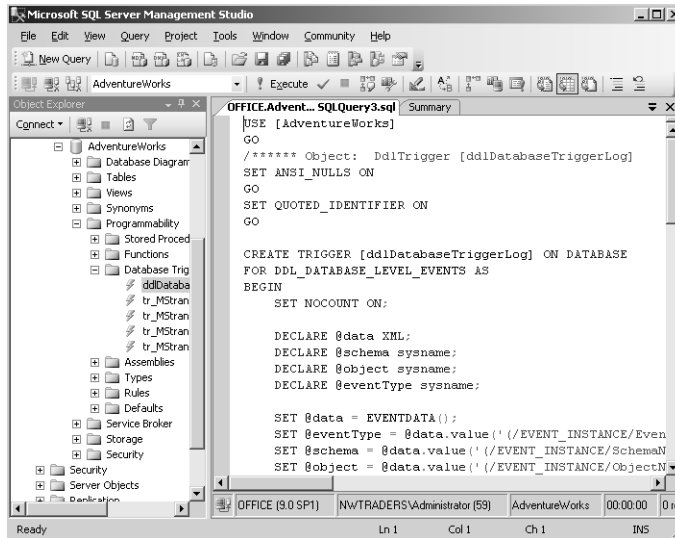


Figure 9-15 Sample trigger in Query Editor.

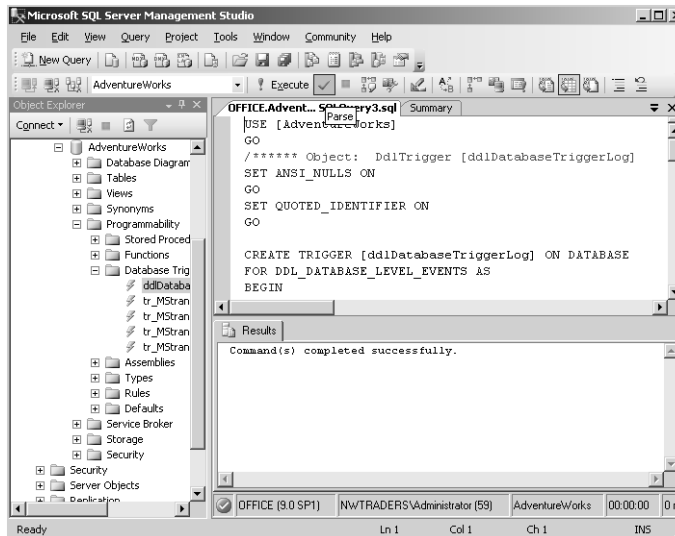


Figure 9-16 Parsing the commands.

9. Click Execute. You are likely to get the message shown in Figure 9-17, because by default the DDL trigger *ddlDatabaseTriggerLog* exists in the AdventureWorks database but is disabled. Microsoft provides the code as an example.

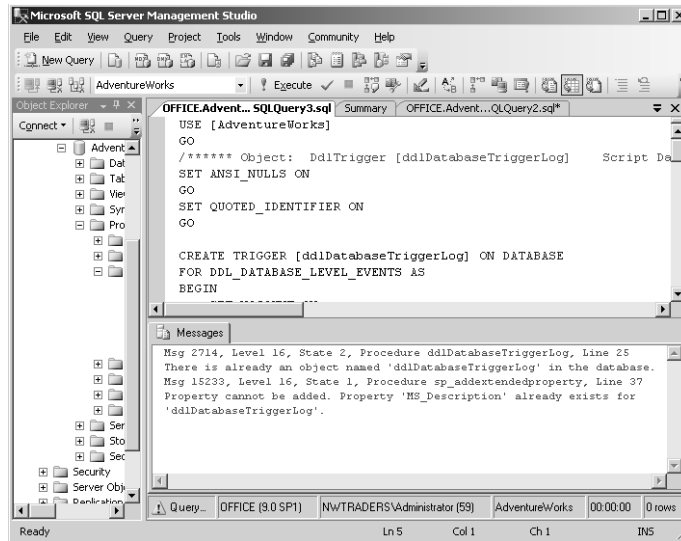


Figure 9-17 Attempting to create the trigger.

Lesson Summary

- You need to implement change control procedures for the various stages of database operations. You can limit ad hoc changes to database structures by using DDL Triggers. You can also audit the use of SQL Server 2005 permissions to help you verify that users with these permissions are observing database change control procedures.
- If you are optimizing a database change control strategy to meet business requirements, you need to identify the database objects related to the particular deployment. If the database is large or if the list of objects changes frequently, using stored procedures can save time and effort.
- All communications between the developers and the DBA team must be implemented either by e-mail or by signed forms. A paper trail, either electronic or physical, must exist at all stages.
- You can create a migration script to convert one database's structure to match that of another.

Lesson Review

You can use the following questions to test your knowledge of the information in Lesson 2, “Optimizing a Database Change Control Strategy to Meet Business Requirements.” The questions are also available on the companion CD if you prefer to review them in electronic form.

NOTE Answers

Answers to these questions and explanations of why each answer choice is correct or incorrect are located in the “Answers” section at the end of the book.

1. One of your company’s subsidiaries submits orders in an XML file that you use as the input argument of a stored procedure. The stored procedure needs to identify any invalid product IDs in the order. One of your company’s databases contains a list of all valid product IDs. You need to ensure that the stored procedure can produce the desired result set. What construct should you use within your stored procedure?
 - A. A SELECT clause with a UNIQUE constraint
 - B. A NOT EXISTS subquery
 - C. A CROSS JOIN statement with a WHERE clause
 - D. Two queries that use the UNION operator
2. One of your company’s SQL Server 2005 databases includes a table named OrderQuantity. Written company policy states that the value in the OrderQuantity column cannot be increased by 5 percent or more during any single database operation. Users can update values in the OrderQuantity column by using ad hoc queries. How do you ensure that company policy is enforced?
 - A. Create a CHECK constraint on the OrderQuantity column.
 - B. Create a DDL trigger that rolls back changes to the OrderQuantity column that violate company policy.
 - C. Create an SQL trace that uses the Audit Database Management event class.
 - D. Create a DML trigger that rolls back changes to the OrderQuantity column that violate company policy.

3. You need to migrate changes in the database structure from a large test database to the equivalent production database. You do not want to lose any data in the production database. You want to perform the task with minimum administrative effort. How should you perform this task?
- A. Use a migration script.
 - B. Delete the old database, and replace it with the new one.
 - C. Rename the test database, and transfer it to the production server. Use a full outer join to merge the two databases. Use the UNIQUE constraint to delete duplicate rows.
 - D. Use SSMS to transfer the database changes.

Chapter Review

To further practice and reinforce the skills you learned in this chapter, you can perform the following tasks:

- Review the chapter summary.
- Review the list of key terms introduced in this chapter.
- Complete the case scenarios. These scenarios set up real-world situations involving the topics of this chapter and ask you to create a solution.
- Complete the suggested practices.
- Take a practice test.

Chapter Summary

- Determining the business requirements for data quality is typically a team effort. However, tools exist (fuzzy logic, data validation, data mining, and so on) that clean dirty data and provide data models and predictions to meet business requirements.
- Microsoft provides a wide variety of tools to enable application developers to ensure that applications enforce data quality. Transact-SQL provides a variety of statements and clauses that you can use to inspect data.
- You can use DDL triggers to control changes to the database structure, and you can audit permissions to determine who is making these changes. You can use stored procedures to identify the database objects related to a particular deployment.
- A paper trail, either electronic or physical, must exist at all stages of database development and maintenance. You can use migration scripts to migrate alterations made and tested at previous stages to your production databases.

Key Terms

Do you know what these key terms mean? You can check your answers by looking up the terms in the glossary at the end of the book.

- canonical row
- checksum

- data mining
- dirty data
- distance function
- fuzzy grouping
- fuzzy matching
- metadata
- mining model
- training

Case Scenarios

In the following case scenarios, you will apply what you've learned about enforcing data quality according to business requirements and implementing a database change control strategy. You can find answers to these questions in the “Answers” section at the end of this book.

Case Scenario 1: Checking and Correcting Invalid Database Entries

You are a senior DBA at Trey Research. Database users at Trey Research enter data into various tables in the Projects database, and run ad hoc queries against this database. Sometimes data is entered incorrectly, in particular project codes, which for historical reasons must adhere to a strict and complex format. The Human Resources (HR), Accounts, and Research Projects departments all make entries to and run queries against the Projects database, and sometimes you need to check this information to find out whether project funding has been correctly allocated or whether personnel allocated to a project actually exist in the HR records. You are also concerned that users could use ad hoc queries to change the structure of the database. Answer the following questions:

1. You need to prevent any structural changes to the Projects database on your production server. In particular, you do not want any CREATE, DROP, or ALTER Transact-SQL statements to run against the database. What should you do?
2. You need to ensure that entries into the ProjectCode column in any of the database tables adhere to the correct format. What construct should you use to enforce this?

3. The HumanResources.Employees table and the ResearchProjects.Staffing table both contain an EmployeeID column. You want to obtain details about who is working on what project, and also a list of research project staff who do not have a corresponding record in the HumanResources.Employees table (probably because their employee identities have been entered incorrectly). How do you go about this?
4. You want to find records in the HumanResources.Employees table that have entries in the EmployeeID column that are similar but not identical to invalid entries in the same column in the ResearchProjects.Staffing table. What would you use to do this?
5. Don Hall recently joined your DBA team. You want to give Don the ability to issue a SELECT statement against every table and view within the Projects database. You do not want to give him administrative or database-owner privileges. To what fixed database role should you add Don's account?

Case Scenario 2: Managing Schema Changes

You are the Senior DBA at TailSpinToys. Database and application developers need to make changes to the TailSpinToys database on a regular basis. You administer a pre-production network, the purpose of which is to test all schema changes before they are migrated to the production SQL Server 2005 servers. Written company policy states that all changes to the database schema must be tested and approved on the pre-production network before they are migrated to the production servers. Answer the following questions:

1. Who is responsible for implementing schema changes on the pre-production and production networks?
2. What should the initial procedure be when developers perceive the need for schema changes?
3. Who is responsible for signing off on changes to the pre-production database schema?
4. What should the procedure be for implementing approved schema changes on the production database?
5. How do you ascertain and prove that the required procedures have been implemented?

Suggested Practices

To help you successfully master the exam objectives presented in this chapter, complete the following tasks.

Enforce Data Quality According to Business Requirements

Complete all the practices in this section.

- **Practice 1: Investigate regulatory requirements.** Investigate the requirements specification for your organization's database structure (or any other to which you have access). Determine if any legislation has required security or schema changes in the databases.
- **Practice 2: Find out more about BizTalk Server 2006.** Investigate BizTalk Server 2006 and the BAM portal. If you have the opportunity, get some hands-on practice in using this software.
- **Practice 3: Use the SQL Server 2005 BI tools.** Practice using SSMS, BDIS, SSIS, SSRS, and SSAS. Build upon the practices and examples given in this and other chapters in this book. In particular, look at how you would build and use data mining models.
- **Practice 4: Create queries to inspect data.** Build upon the short query examples in the section "Creating Queries to Inspect the Data" in this chapter. Ensure that you fully understand the statements and constructs contained in these examples.

Optimize a Database Change Control Strategy to Meet Business Requirements

Complete all the practices in this section.

- **Practice 1: Investigate source control software.** Find out more about source control software, in particular the VSS package.
- **Practice 2: Use DDL triggers.** Investigate DDL triggers and build on the short example in the practice session "Creating, Testing, and Removing a DDL Trigger" in this lesson.
- **Practice 3: Create stored procedures.** Create stored procedures, building upon the short example in the section "Using Stored Procedures" in this lesson.

- **Practice 4: Investigate migration packages.** Investigate the use of migration software, and find out more about the third-party tools identified in this lesson. Find out if other packages are available for this purpose.

Take a Practice Test

The practice tests on this book's companion CD offer many options. For example, you can test yourself on just one exam objective, or you can test yourself on all the 70-444 certification exam content. You can set up the test so that it closely simulates the experience of taking a certification exam, or you can set it up in study mode so that you can look at the correct answers and explanations after you answer each question.

MORE INFO Practice tests

For details about all the practice test options available, see the "How to Use the Practice Tests" section in this book's Introduction.

Chapter 10

Replication

Microsoft provides a set of replication technologies for copying and distributing data and database objects from one database to another and for synchronizing between databases to maintain consistency. You can use replication to distribute data to different locations and to remote or mobile users over local and wide area networks (WANs), dial-up connections, wireless connections, and the Internet. Microsoft SQL Server 2005 uses a publishing industry metaphor to describe the components of the replication system, which include the Distributor, the Publisher, Subscribers, articles, publications, and subscriptions.

Exam objectives in this chapter:

- Design a strategy to manage replication.
 - Design alerts.
 - Design a maintenance plan to monitor health, latency, and failures.
 - Verify replication.
 - Design a plan to resolve replication conflicts.
 - Design a plan to modify agent profiles.
 - Tune replication configuration.

Lessons in this chapter:

- Lesson 1: Designing a Strategy to Manage Replication 525

Before You Begin

To complete the lesson in this chapter, you must have done the following:

- Configured a Microsoft Windows Server 2003 R2 computer with SQL Server 2005 Enterprise Edition SP1 as detailed in the Appendix.
- Installed an updated copy of the AdventureWorks sample database as detailed in the Appendix.

No additional configuration is required for this chapter.

Real World

Ian McLean

Because replication is relatively straightforward to configure, a tendency exists to use it for almost all data transfers and for disaster recovery. Suppose, for example, you are replicating a database between two servers with a replication latency measured in minutes and you back up the database every night and the transaction logs every two hours. If the database on one of the servers becomes corrupt but the second server is unaffected, the fastest and easiest way to recover probably is to back up the database on the second server and restore it to the first.

However, replication uses resources and is not typically the best disaster recovery solution. You should use replication to synchronize changes to remote databases with a central database, to create multiple instances of a database and distribute the workload, to distribute data that updates on a regular basis from a central location to other servers, and to customize data and distribute it to other Subscribers—and sometimes as part of a failover strategy.

When I first came across *database mirroring*, it was disabled by default, and Microsoft advised users to enable it only for validation on test networks. I validated database mirroring on my own test network and decided that if I needed to provide fault tolerance for a database and ensure that client computers can connect to the database—even if the SQL Server 2005 computer experiences a complete hardware failure—with minimal administrative effort and disruption to the user, I would mirror the database on a second computer just as soon as Microsoft told me it was OK to do so. So when SP1 was released, I installed the service pack on my test network, checked it out, checked that database mirroring still worked OK, and then installed SP1 and implemented database mirroring on a production network.

I suppose one moral of this story is that test networks are invaluable. However, the main point is that replication has its place, as has clustering, backup and restore, log shipping, and database mirroring. I discuss the use of replication with database mirroring later in this chapter. Each tool has its specific purpose, though these tools might overlap. No one tool fits all.

Lesson 1: Designing a Strategy to Manage Replication

A database administrator (DBA) needs to know how to configure replication. However, a number of high-level tasks need to be tackled before you start your configuration. You need to formulate the replication strategy and design the replication model that is appropriate to your enterprise. You need to decide what you are going to replicate, when you are going to replicate, and which servers you intend to use to distribute or receive data (or both).

Nor is your job finished when you have implemented your replication plan and checked that it works. You also need a plan to monitor replication and ensure that it continues to operate at the efficiency levels you first specified. You need to configure alerts and warnings that let you know about any problems long before they become apparent to your users. You need to configure your *replication agents* to work at maximum efficiency by generating and configuring agent profiles. Tuning replication configuration is a continuous process, and you can always find a means of significantly improving efficiency if you look hard enough.

In this chapter, you learn how to design strategies to configure alerts, monitor replication health, monitor latency and failures, verify and tune replication, resolve conflicts, and configure agent profiles. The chapter also discusses database mirroring and how this new feature works with replication.

NOTE Service packs

The service pack level at the time of this writing is Service Pack 1 (SP1). Unless otherwise indicated, all the information in the chapter applies to both SQL Server 2005 and SQL Server 2005 SP1.

After this lesson, you will be able to:

- List the various types of replication and replication topologies that SQL Server 2005 uses and select the appropriate type for a given scenario.
- Design an alert strategy, and configure replication alerts.
- Design a maintenance strategy that monitors the replication process and reports on latency and failures.
- Verify replication by using *tracer tokens* and *checksum*.
- Resolve replication conflicts, modify agent profiles, and tune replication configuration.
- Discuss the advantages of using replication with database mirroring.

Estimated lesson time: 75 minutes

Selecting a Replication Strategy

Replication falls into two broad categories: replicating data in a server-to-server environment, and replicating data between a server and clients.

Server-to-Server Replication

Data is typically replicated between servers to provide the following features:

- **Improved scalability and availability** You can use replication to maintain continuously updated copies of data and to spread read activity across multiple servers. The redundancy that results from maintaining multiple copies of the same data permits failover support during planned and system maintenance, or if a server fails.
- **Data warehousing and reporting** Data warehouse and reporting servers typically use data from online transaction processing (OLTP) servers. You can use replication to move data between OLTP servers and reporting and decision support systems.
- **Integrating data from multiple sites** You can replicate data from remote offices and consolidate it at a central office. You can then replicate the consolidated data to the remote offices.
- **Integrating heterogeneous data** Some applications send data to or obtain data from databases other than SQL Server 2005 databases. You can use replication to integrate data from non-SQL Server databases.
- **Offloading batch processing** Batch operations can be resource intensive, and you should not run them on an OLTP server. You can use replication to transfer the data to a dedicated batch processing server.

Replication Between Servers and Clients

You can replicate data between servers and clients, including workstations, laptops, tablets, and devices. Typically, you replicate data between servers and clients to support the following features:

- **Exchanging data with mobile users** Many applications require you to make data available to remote users, including sales personnel and delivery drivers. These applications include Customer Relationship Management (CRM), sales force automation (SFA), and field force automation (FFA) applications.
- **Obtaining data from consumer point-of-sale (POS) applications** POS applications, such as checkout terminals and automatic teller machines (ATMs), require that you replicate data from remote sites to a central site.

- **Integrating data from multiple sites** Many applications integrate data from multiple sites. For example, an application that supports regional offices might require data to flow in one or both directions between regional offices and a central office. This feature can be implemented by either server-to-server replication or replication between a server and clients.

Specifying a Replication Type, Topology, and Model

SQL Server 2005 provides the following types of replication for use in distributed applications:

- Transactional replication
- Merge replication
- Snapshot replication

The type of replication you choose for an application depends on many factors—for example, the physical replication environment, the type and quantity of data to be replicated, and whether the data is updated at the Subscriber. The number and location of computers involved in replication and whether these computers are clients or servers defines the physical environment.

Each type of replication typically begins with an initial synchronization of the published objects between the Publisher and Subscribers. Replication with a snapshot, which is a copy of all the objects and data specified by a publication, performs this initial synchronization. After the snapshot is created, it is delivered to the Subscribers. For some applications, only snapshot replication is required. Other types of applications require subsequent data changes to flow to the Subscriber incrementally over time. Some applications also require that changes flow from the Subscriber back to the Publisher. Peer-to-peer transactional replication, transactional replication with updating subscriptions, and merge replication provide options for these types of applications.

Data changes are not tracked for snapshot replication; each time a snapshot is applied, it completely overwrites the existing data. Transactional replication tracks changes through the SQL Server transaction log, and merge replication tracks changes through triggers and metadata tables.

Transactional Replication

Transactional replication typically starts with a snapshot of the publication database objects and data. As soon as the initial snapshot is taken, subsequent data changes and schema modifications made at the Publisher are typically delivered to the

Subscriber as they occur. Data changes are applied to the Subscriber in the same order and within the same transaction boundaries as they occur at the Publisher. This guarantees transactional consistency within a publication.

You typically use transactional replication in server-to-server environments and it is appropriate in each of the following cases:

- You want SQL Server to propagate incremental changes to Subscribers as they occur.
- The application requires low *latency* (the amount of time that elapses between a transaction being committed at the Publisher and the corresponding transaction being committed at the Subscriber).
- The application requires access to intermediate data states. For example, if a row changes several times, transactional replication allows an application to respond to each change (by, for example, firing a trigger) rather than implement an update only once at the Publisher to reflect the net data change.
- The Publisher has a high volume of insert, update, and delete activity.
- The Publisher or Subscriber is a non-SQL Server database, such as Oracle.

MORE INFO Allowing updates at the Subscriber

By default, Subscribers to transactional publications should be treated as read-only, because changes are not propagated back to the Publisher. However, transactional replication does offer options that allow updates at the Subscriber. For more information, search for "Publication Types for Transactional Replication" in Books Online or access [msdn2.microsoft.com/en-us/library/ms152570\(d=ide\).aspx](http://msdn2.microsoft.com/en-us/library/ms152570(d=ide).aspx).

Merge Replication

Merge replication typically starts with a snapshot of the publication database objects and data. Subsequent data changes and schema modifications made at the Publisher and Subscribers are tracked by using triggers. The Subscriber synchronizes with the Publisher and exchanges all rows that have changed between the Publisher and Subscriber since the last time synchronization occurred.

Merge replication is typically used in server-to-client environments and is appropriate in any of the following situations:

- Multiple Subscribers might update the same data at various times and propagate those changes to the Publisher and to other Subscribers.

- Subscribers need to receive data, make changes offline, and later synchronize changes with the Publisher and other Subscribers.
- Each Subscriber requires a different partition of data.
- You need to be able to detect and resolve conflicts.
- The application requires net data change rather than access to intermediate data states. For example, if a row changes several times at a Subscriber before it synchronizes with a Publisher, the row will change only once at the Publisher to reflect the net data change (that is, the final value).

Merge replication allows various sites to work autonomously and subsequently to merge updates into a single, uniform result. Because updates are made at more than one node, the same data might have been updated by the Publisher and by more than one Subscriber. Therefore, conflicts can occur when updates are merged, and merge replication provides a number of ways to handle conflicts.

MORE INFO Detecting and resolving merge replication conflicts

For more information about how merge replication detects and resolves conflicts, search for "Detecting and Resolving Merge Replication Conflicts" in Books Online or access [msdn2.microsoft.com/en-us/library/ms151191\(d=ide\).aspx](http://msdn2.microsoft.com/en-us/library/ms151191(d=ide).aspx).

Using Triggers, Constraints, and NOT FOR REPLICATION A *trigger* is a type of stored procedure that automatically takes effect when a language event executes. When a data definition language (DDL) event takes place in a server or database, it can invoke a DDL trigger (introduced in SQL Server 2005). When a data manipulation language (DML) event takes place in the database, it can invoke a DML trigger.

DML events include INSERT, UPDATE, or DELETE statements that modify data in a specified table or view. A DML trigger can query other tables and can include complex Transact-SQL statements. The trigger and the statement that fires it are treated as a single transaction, which can roll back from within the trigger.

Because of SQL Server 2005 integration with the .NET Framework common language runtime (CLR), you can use any .NET Framework language to create CLR triggers. You can, for example, create a trigger that prints an invoice when a new sale is entered into a sales database.

You need to take care to configure triggers appropriately when you are setting up replication. Suppose, for example, that your organization has several sales outlets, each with its own SQL Server 2005 server, that replicate sales data with a central office

by using merge replication, and each outlet has implemented a trigger that initiates an invoice for local sales. You want to replicate sales data, but you do not want the trigger to operate on data from other outlets that replicates to an outlet from the central office. You would therefore configure the trigger on the server in each sales outlet to use the NOT FOR REPLICATION (NFR) clause. The NOT FOR REPLICATION clause stops SQL Server from executing the trigger when a replication agent modifies the table upon which the trigger is configured. It does not stop the trigger from being published as a part of SQL Server replication. If you do not want to replicate a trigger, you can use the WITH ENCRYPTION clause.

You can use CHECK constraints to prevent users from entering data in an incorrect format. For example, if a product code is always six characters long, a CHECK constraint can prevent a user from entering a code of the incorrect length. If you do not want to replicate constraints, you can use the NOT FOR REPLICATION clause on the table on the Publisher that contains the constraint. A FOREIGN KEY constraint allows certain fields in one table to refer to fields in another table. You can also use the NOT FOR REPLICATION clause to prevent the replication of FOREIGN KEY constraints. Notice that the NOT FOR REPLICATION clause limits the operation of triggers but does not prevent their replication. However, it does prevent the replication of constraints.

MORE INFO NOT FOR REPLICATION clause

For more information about the NOT FOR REPLICATION clause, search for "Controlling Constraints, Identities, and Triggers with NOT FOR REPLICATION" in Books Online or access [msdn2.microsoft.com/en-us/library/ms152529\(d=ide\).aspx](http://msdn2.microsoft.com/en-us/library/ms152529(d=ide).aspx).

Snapshot Replication

Snapshot replication distributes data as it appears at a specific moment in time and does not monitor data updates. When synchronization occurs, the entire snapshot is generated and sent to Subscribers.

IMPORTANT The snapshot process is part of transactional and merge replication.

Snapshot replication can be used by itself, but the snapshot process (which creates a copy of all the objects and data specified by a publication) also provides the initial set of data and database objects for transactional and merge publications.

Using snapshot replication by itself is most appropriate when one or more of the following is true:

- Data changes infrequently.
- You can accept that copies of data are out of date with respect to the Publisher for a period of time.
- You are replicating small volumes of data.
- A large volume of changes occurs over a short period of time.

Snapshot replication is most appropriate when data changes are substantial but infrequent. For example, if an organization maintains a product catalog that is updated once per year, replicating the entire snapshot of the catalog data after the update has been completed is the recommended procedure. Snapshot replication is also used if changes to a database are made at the Publisher during the day and are then replicated to Subscribers at night.

Snapshot replication has a lower continuous overhead on the Publisher than transactional replication, because incremental changes are not tracked. However, if you are replicating a large data set, snapshot replication requires substantial resources to generate and apply the snapshot. You need to consider the size of the entire data set and the frequency of changes to the data when you are deciding whether to use snapshot replication.

MORE INFO Implementing replication

For more information about implementing transactional, merge, and snapshot replication, search for “Implementing Replication” in Books Online or access [msdn2.microsoft.com/en-us/library/ms151847\(d=ide\).aspx](https://msdn2.microsoft.com/en-us/library/ms151847(d=ide).aspx).

Replication Topology

When you configure database replication—for example, by using Microsoft SQL Server Management Studio (SSMS) and the Configure Distribution Wizard—you set up a topology based on the replication model you are implementing. A replication topology contains objects that carry out the following roles:

- **Distributor** The machine that distributes data. The Distributor makes the publication available to the Subscriber. It creates tables and files to store the data it replicates, and you need to choose a machine that has sufficient disk space to do so. Optionally, you can configure a Distributor to store a snapshot replication file

at another location (the Snapshot Offline option). The simplest replication model places the Distributor and Publisher on the same machine, but moving the Distributor function to its own server can provide advantages by, for example, addressing disk usage and security issues.

- **Publisher** The machine that stores the data you want to replicate. This server takes the heaviest load, depending on the type of replication you select. For example, snapshot replication that is scheduled for off-peak hours has a smaller continuous impact than constant transactional replication.
- **Subscriber** The database that receives the data. Subscribers need to be able to access the Distributor, unless you configure the Snapshot Offline option. Subscriber configuration depends on the type of replication you choose. For example, push-snapshot replication requires a different Subscriber configuration than merge-pull replication.

When you have decided which nodes (machines or devices) host the replication objects, you next need to consider the data that you want to replicate. You structure this data by using the following objects:

- **Article** The base-level replication object. Articles can, for example, include the following:
 - Tables
 - Stored procedures
 - Views
 - Indexed views
 - User-defined functions
- **Publication** Contains the articles you want to send. You could, for example, select a table as one article, an indexed view as another, and then combine those articles as a publication. The Subscriber always subscribes to a publication, not an article. Even if you select only one article, you need to place it in a publication to transfer the data.
- **Subscription** The function that receives the publication. This is typically another SQL Server 2005 server, but it can include client workstations or devices. You can also replicate to non-SQL Server databases—for example, Oracle and Microsoft Access.

IMPORTANT Publish all referenced objects.

If you are publishing a database object that depends on other database objects, you must publish all referenced objects. For example, if you publish a view that depends on a table, you must also publish the table.

Replication Models

A replication model is a map of how data is distributed across your SQL Server 2005 network and how you configure your servers during replication implementation. When you choose a replication model, you need to bear in mind the physical layout of your Publisher, Distributor, and Subscriber databases. You can implement one (or more) of the following models:

- **Peer-to-Peer** Allows replication between identical participants in the topology. This model enables roles to move dynamically between nodes for the purposes of maintenance and failure management. The disadvantage of the Peer-to-Peer model is the additional administrative overhead involved with moving roles.
- **Central Publisher** Maintains the Publisher and Distributor roles on the same SQL Server 2005 server, with one or more Subscriber roles configured on other servers. This model provides ease of maintenance and management but places a significant additional workload on the publication server.
- **Central Publisher with Remote Distributor** Implements the Publisher and Distributor roles on separate SQL Server 2005 servers, with one or more Subscriber roles configured on other servers. The workload is more evenly distributed than on the Central Publisher model, but you need to maintain an additional server.
- **Central Subscriber** Provides a Subscriber role on a single server that collects information from several Publishers. The Central Subscriber collates the information, and you can optionally configure it to then republish the collated information to other servers. If you use this replication model, you need to ensure that all tables used in replication have a unique primary key.
- **Publishing Subscriber** Obtains data from a Publisher and relays it to other Subscribers. You typically use this model with one of the other models. For example, you could configure a Publisher to replicate to a publishing Subscriber at a remote location. The publishing Subscriber can then replicate the information to other Subscribers at its location.

Designing and Configuring Replication Alerts

If you have chosen the appropriate replication model, ensured that your servers have sufficient resources to cope with their assigned roles (for now and in the foreseeable future), and configured replication correctly, the process should occur without user intervention. However, this does not mean that you can ignore it. You need to monitor the replication process, and you need to configure alerts to warn you when anything goes wrong.

Predefined Alerts

SSMS and Microsoft SQL Server Agent enable you to use alerts to monitor events—for example, replication agent events. SQL Server Agent monitors the Windows application log for events that are associated with alerts. If such an event occurs, SQL Server Agent responds by executing a predefined task, sending an e-mail or a pager message to a specified DBA, or both. SQL Server 2005 provides a set of predefined replication alerts for replication agents. Table 10-1 lists the predefined alerts that SQL Server 2005 installs when a computer is configured as a Distributor.

Table 10-1 Predefined Replication Alerts

Predefined Alert	Condition That Causes the Alert to Fire	Message Identity
Replication: agent success	The replication agent shuts down successfully.	14150
Replication: agent failure	The replication agent shuts down with an error.	14151
Replication: agent retry	The replication agent shuts down after unsuccessfully retrying an operation. Typically, this happens when the agent encounters an error such as “server not available,” “deadlock,” “connection failure,” or “time-out failure.”	14152
Replication: expired subscription dropped	An expired subscription is dropped.	14157
Replication: Subscription reinitialized after validation failure	The response job “Reinitialize subscriptions on data validation failure” reinitializes a subscription successfully.	20572

Table 10-1 Predefined Replication Alerts

Predefined Alert	Condition That Causes the Alert to Fire	Message Identity
Replication: subscriber has failed data validation	The Replication Distribution Agent (distrib.exe) or the Replication Merge Agent (replmerg.exe) fails data validation.	20574
Replication: subscriber has passed data validation	The Replication Distribution Agent or the Replication Merge Agent passes data validation.	20575
Replication: agent custom shutdown	The replication agent shuts down for a predefined reason.	20578

You can configure these alerts from the Alerts folder in SSMS or the Warnings And Agents tab in Microsoft SQL Server Replication Monitor.

MORE INFO Replication Distribution Agent and Replication Merge Agent

For more information about these executables, search for “Replication Distribution Agent” and “Replication Merge Agent” in Books Online, or access msdn2.microsoft.com/en-us/library/ms147328.aspx and [msdn2.microsoft.com/en-us/library/ms147839\(d=ide\).aspx](http://msdn2.microsoft.com/en-us/library/ms147839(d=ide).aspx).

SQL Server 2005 replication provides a response job for subscriptions that fail data validation, and a framework for creating additional automated responses to alerts. The response job “Reinitialize subscriptions on data validation failure” is stored in the SQL Server Agent Jobs folder in SSMS. If articles in a transactional publication fail validation, the response job reinitializes only those articles that failed. If articles in a merge publication fail validation, the response job reinitializes all articles in the publication.

Typically when an alert triggers, the only information it provides is contained in the alert message itself. Parsing this information can be difficult. SQL Server 2005 replication makes it easier to automate responses by providing additional information about the alert in the *sysreplactionalerts* system table, which is stored in the msdb database. This table provides parsed information in a form that customized programs can use.

Replication Monitor Warnings

In addition to the predefined replication alerts described in the previous section, Replication Monitor provides warnings that you can configure to alert you when certain conditions occur. Replication Monitor displays status information for publications and subscriptions, and by default it displays warnings only for uninitialized subscriptions. However, you can enable warnings for other conditions to enable you to obtain information about status and performance in a timely manner.

You can enable a warning and, if relevant, specify a threshold. In addition to displaying a warning in Replication Monitor, you can also configure an alert. You can enable warnings for the following conditions:

- **Imminent subscription expiration** Applies to all types of replication. If the specified threshold is met or exceeded, the subscription status is displayed as *Expiring soon*.
- **Exceeding the specified latency** Applies to transactional replication. If the specified threshold is met or exceeded, the subscription status is displayed as *Performance critical*.
- **Exceeding the specified synchronization time** Applies to merge replication. If the specified threshold is met or exceeded, the status is displayed as *Long-running merge*. You can specify different thresholds for dial-up and local area network (LAN) connections.
- **Falling short of processing the specified number of rows in a given amount of time** Applies to merge replication. If the specified threshold is met or exceeded, the status is displayed as *Performance critical*. You can specify different thresholds for dial-up and LAN connections.

MORE INFO Setting thresholds and warnings, and configuring alerts

For more information about setting thresholds and warnings and configuring alerts for the various replication types, search for “Setting Thresholds and Warnings in Replication Monitor” in Books Online and click the links “How to: Set Thresholds and Warnings for a Transactional Publication (Replication Monitor),” “How to: Set Thresholds and Warnings for a Merge Publication (Replication Monitor),” and “How to: Set Thresholds and Warnings for a Snapshot Publication (Replication Monitor).” Alternatively, access [msdn2.microsoft.com/en-us/library/ms152521\(d=ide\).aspx](http://msdn2.microsoft.com/en-us/library/ms152521(d=ide).aspx), [msdn2.microsoft.com/en-us/library/ms151173\(d=ide\).aspx](http://msdn2.microsoft.com/en-us/library/ms151173(d=ide).aspx), and [msdn2.microsoft.com/en-us/library/ms152575\(d=ide\).aspx](http://msdn2.microsoft.com/en-us/library/ms152575(d=ide).aspx).

You can select the following specific warnings for each replication type:

- Merge publication warnings
 - Warn when a subscription will expire within the threshold.

- ❑ Warn when a merge length for dial-up connections exceeds the threshold.
- ❑ Warn when a merge length for LAN connections exceeds the threshold.
- ❑ Warn when rows merged per second for LAN connections is less than the threshold.
- ❑ Warn when rows merged per second for dial-up connections is less than the threshold.
- Transactional publication warnings
 - ❑ Warn when a subscription will expire within the threshold.
 - ❑ Warn when latency exceeds the threshold.
- Snapshot publication warnings
 - ❑ Warn when a subscription will expire within the threshold.

Monitoring Replication Status

You can use Replication Monitor to monitor the overall health of a replication topology, identify slow subscriptions, and measure latency. The tool also enables you to monitor failures caused by agents that are not running when they should be.

To monitor replication, you must be a member of the *sysadmin* fixed server role at the Distributor or (at least) a member of the *replmonitor* fixed database role in the distribution database. If you want to enable other users to view replication activity but not to administer replication, you can add their accounts to the *replmonitor* role.

Replication Monitor presents a view of replication activity in a two-pane format, as shown in Figure 10-1. You add a Publisher to the monitor in the left pane, and in the right pane the monitor displays information about the Publisher, its publications, subscriptions to those publications, and the various replication agents. Replication Monitor also enables you to start and stop agents and validate data.

Monitoring Replication Health

The replication system is relatively healthy if there are no error icons on nodes in the left pane. You can also check the Subscription Watch List tab, which displays information about subscriptions that might require attention. Figure 10-2 shows the Subscription Watch List tab.

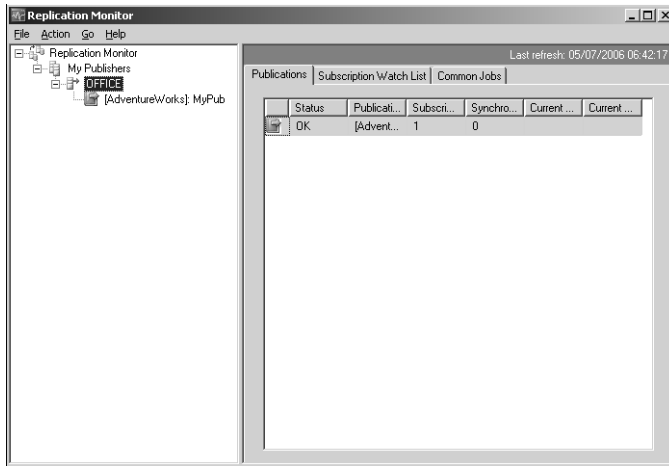


Figure 10-1 Replication Monitor.

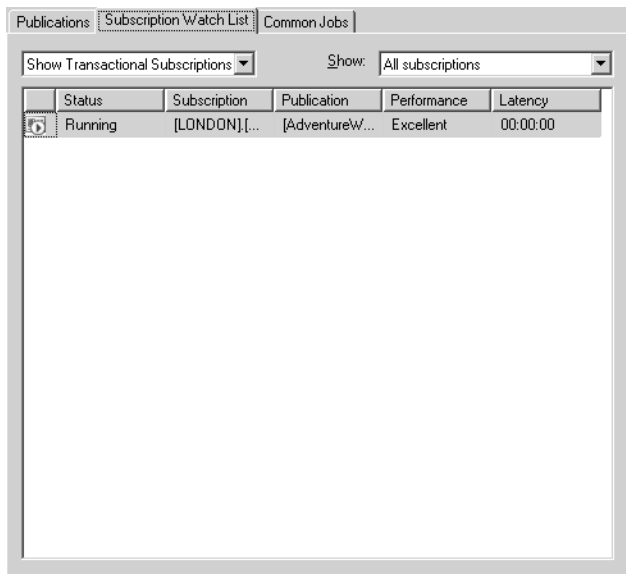


Figure 10-2 The Subscription Watch List tab.

Monitoring Agents

If an agent is not running, either it is not scheduled to run or an error has occurred. In the latter case, Replication Monitor displays an error icon on the appropriate nodes in the left pane. For example, if the Snapshot Agent for a publication stopped because of an error, an error icon is displayed on the Publisher Group, Publisher, and Publication nodes.

Monitoring Publisher Information

Replication Monitor displays information about Publishers on the three following tabs:

- **Publications** Provides summary information for all publications at a Publisher.
- **Subscription Watch List** Displays information about subscriptions from all publications available at the selected Publisher. You can filter the list of subscriptions to display errors, warnings, and poorly performing subscriptions. You can use the tab to monitor replication health by displaying error and warning icons for any subscriptions that require your attention. The tab also enables you to access subscription properties, access detailed information about the agent or agents associated with a subscription, reinitialize subscriptions, and validate subscriptions.
- **Common Jobs** Displays information about the jobs used by all the replication types. The tab also allows you to access detailed information about the jobs and start and stop each job.

Replication Monitor also provides a context menu for the Publisher node. You can right-click a Publisher in the left pane and choose one of the following options:

- Add a Publisher to Replication Monitor.
- Edit Replication Monitor settings for the Publisher.
- Remove the Publisher from Replication Monitor.
- Connect to or disconnect from the Distributor that stores information about the Publisher.
- View and edit agent profiles.
- Configure replication alerts.

Monitoring Publication Information

Monitoring publication information provides another method of monitoring replication health and identifying poorly performing subscriptions. It also enables you to display information about the agents associated with the publication and to set thresholds and warnings. The section “Designing and Configuring Replication Alerts” earlier in this lesson discusses this functionality in detail. Finally, this functionality enables you to measure latency for transactional replication. Replication Monitor displays

information about publications on the three following tabs and a number of detail windows:

- **All Subscriptions** Displays information about all subscriptions to the selected publication. By default, the subscriptions appear in priority order: errors and then warnings. Then they appear in order of performance, with any poorly performing subscriptions at the top. The errors and warnings assist you in monitoring replication health, and the order of the list enables you to identify poorly performing subscriptions.
- **Warnings and Agents** Displays information about all agents associated with the publication, and allows you to specify warnings and alerts.
- **Tracer Tokens** Enables you to measure latency in transactional replication. Tracer tokens indicate how long it takes for a transaction committed at a particular time to reach a Subscriber, and then the tokens compare this value with previous times.

NOTE Tracer tokens

The tracer tokens facility in Replication Monitor uses tracer tokens to measure latency. A *tracer token* is a record that a Distributor can send to a Subscriber. Replication Monitor, or a Transact-SQL procedure, can then use the identity of the tracer token (*@tracer_id*) to indicate replication latency. The *sys.sp_posttracertoken* stored procedure posts the tracer token, the *sys.sp_helptracertokens* stored procedure calculates latency, and the *sys.sp_helptracertokenhistory* stored procedure displays tracer token history.

Replication Monitor also provides detail windows for the agents associated with a publication. The following agents are associated with publications:

- **Snapshot Agent** Used by all publications
- **Log Reader Agent** Used by all transactional publications
- **Queue Reader Agent** Used by transactional publications enabled for queued updating subscriptions

Figure 10-3 shows the Warnings And Agents tab and the context menu for the Snapshot Agent.

In addition to the agents associated with publications, Replication Monitor provides information about the following agents associated with subscriptions:

- **The Distribution Agent** Used for subscriptions to snapshot and transactional publications

- **The Merge Agent** Used for subscriptions to merge publications

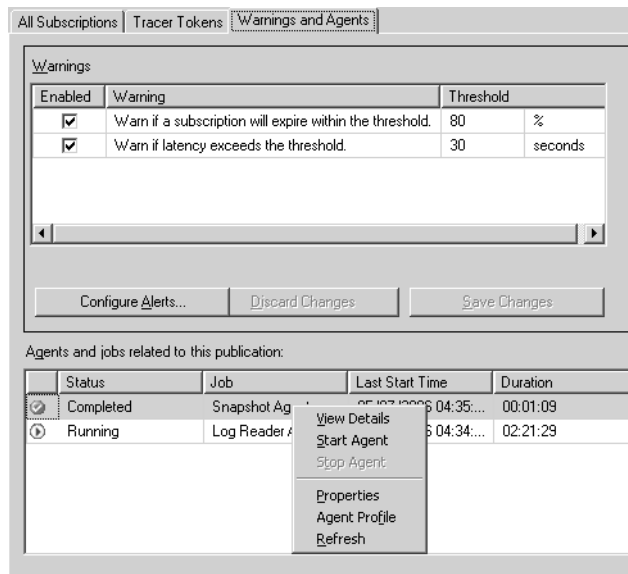


Figure 10-3 The Warnings And Agents tab.

You can double-click an agent to access information in a detail window. The agent detail windows provide information about agent sessions, including start time, end time, duration, and the actions performed in a session. The error messages provide detailed information about why an agent is not running, and they assist you in troubleshooting issues with the agents associated with a publication.

Replication Monitor also provides a context menu for the publications node. You can right-click a publication in the left pane to choose one of the following tasks:

- Reinitialize all subscriptions to a publication.
- Validate all subscriptions to a publication.
- Generate a snapshot for a publication.
- View and edit publication properties.

Monitoring Subscription Information

Replication Monitor displays information about subscriptions on several different tabs. You double-click a subscription in Replication Monitor to access these tabs in a detail window. Error messages on the tabs provide detailed information about why an

agent is not running, and they provide a starting point for troubleshooting issues with the agents associated with a subscription. The following tabs are related to subscriptions:

- **All Subscriptions** Described earlier in this section.
- **Subscription Watch List** Described earlier in this section.
- **Publisher To Distributor History** Used for transactional replication only. This tab displays information about the Log Reader Agent for a publication. The tab is identical to the Log Reader Agent details window.
- **Distributor To Subscriber History** Used for both snapshot replication and transactional replication. This tab displays information about the Distribution Agent for a subscription.
- **Undistributed Commands** Used for transactional replication only. This tab displays information about the number of commands in the distribution database that have not been delivered to the selected Subscriber, and the estimated time to deliver those commands.
- **Synchronization History** Used for merge replication only. This tab displays information about the Merge Agent for a subscription. The information on this tab helps you to identify problems that are slowing down merge replication. The tab provides detailed statistics for each article processed during synchronization, including the amount of time required by each processing phase. It can help pinpoint specific tables that are causing slowdowns, and it provides the best tool to troubleshoot performance issues with merge subscriptions.

MORE INFO Viewing publication and subscription status in Replication Monitor

For more information, and in particular to obtain a list of the icons Replication Monitor uses to indicate publication and subscription values, search for "Viewing Publication and Subscription Status in Replication Monitor" in Books Online or access [msdn2.microsoft.com/en-us/library/ms151271\(d=ide\).aspx](http://msdn2.microsoft.com/en-us/library/ms151271(d=ide).aspx).

Viewing Performance Measurements

Replication Monitor displays performance quality values for transactional and merge replication in the Current Average Performance and Current Worst Performance columns for publications and in the Performance column for subscriptions. The values are as follows:

- Excellent
- Good
- Fair

- Poor
- Critical (transactional replication only)

For transactional replication, performance quality is determined by the latency threshold. If the threshold is not set, a value is not displayed. Table 10-2 shows the correlation between the threshold and the performance quality value. For example, if the threshold is set to 60 seconds and the actual latency is 45 seconds, latency is 75 percent of the threshold, resulting in a value of Fair.

Table 10-2 Correlation Between Threshold and Performance Quality Value

Excellent	Good	Fair	Poor	Critical
0–34%	35–59%	60–84%	85–99%	100%+

For merge replication, performance is determined by comparing individual subscription performance to the average historical performance of subscriptions to the publication that have the same connection type (dial-up or LAN). Replication Monitor displays a value after five synchronizations have occurred with 50 or more changes each over the same type of connection. If there have been fewer than five synchronizations with 50 or more changes, or if the most recent synchronization has fewer than 50 changes, Replication Monitor does not display a value.

Table 10-3 shows the correlation between the average performance and the performance quality value. For example, if 10 Subscribers have previously synchronized over a LAN connection at an average rate of 100 rows per second, and a subscription subsequently synchronizes at a rate of 125 rows per second, the performance of that Subscriber's synchronization is 125 percent of the average, resulting in a value of Good.

Table 10-3 Correlation Between Average Performance and Performance Quality Value

Excellent	Good	Fair	Poor
151+ %	76–150%	26–75%	0–25%

Verifying Replication

To verify replication, you need first to determine that it is implemented across your enterprise, and that every server that should receive data does so. Secondly, you need to validate the data that your Subscribers receive, and ensure that no data becomes corrupt during the replication process.

Using Tracer Tokens

Transactional replication provides the tracer token feature, which gives you a method of measuring latency in transactional replication topologies and validating the connections between the Publisher, Distributor, and Subscribers. A token is a small amount of data that is written to the transaction log of the publication database, marked as though it were a typical replicated transaction, and sent through the system to enable Replication Monitor to calculate the following:

- The amount of time that elapses between a transaction being committed at the Publisher and the corresponding command being inserted in the distribution database at the Distributor.
- The amount of time that elapses between a command being inserted in the distribution database and the corresponding transaction being committed at a Subscriber.

From these calculations, you can identify the Subscribers that take the longest time to receive a change from the Publisher, and any that do not receive an expected tracer token at all.

Subscriptions must be active to receive a tracer token. A subscription is active if it has been initialized. Reinitialization removes any pending tracer tokens for the relevant subscriptions, and Subscribers receive only tracer tokens that were created after their initial synchronization. Tracer tokens are not forwarded by republishing Subscribers.

Tracer tokens can also be useful when *quiescing* a system. This process involves stopping all activity and verifying that all nodes have received all outstanding changes.

Validating Replicated Data

If you have configured transactional or merge replication, you can validate that data at the Subscriber matches data at the Publisher. You can perform validation for specific subscriptions or for all subscriptions to a publication by right-clicking the publication in Replication Monitor and choosing Validate Subscriptions. You can click Validation Options in the Validate Subscriptions dialog box and specify one of the validation options shown in Figure 10-4, and either the Distribution Agent or Merge Agent validates data the next time the agent runs.

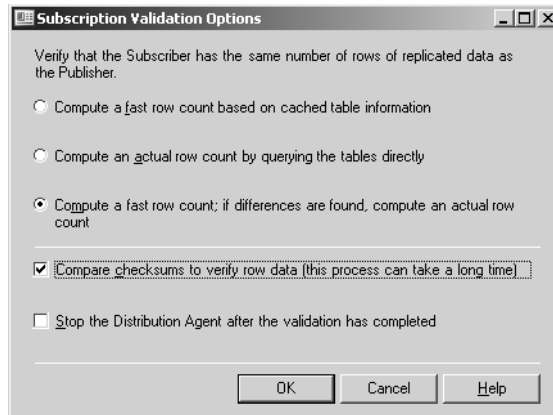


Figure 10-4 Subscription validation options.

NOTE Checksum algorithm

SQL Server 2005 calculates the checksum by using the binary checksum algorithm introduced in Microsoft SQL Server 2000.

Using the Merge Agent

If you have specified merge replication, you can validate that data is partitioned correctly for each Subscriber in addition to validating that data at the Subscriber and Publisher match. If you enable validation, the Merge Agent validates the information at the Subscriber and ensures that each Subscriber's partition is the same as the one received in the initial snapshot whenever the Subscriber reconnects to the Publisher for the next synchronization. For each subsequent merge or snapshot application, the Merge Agent validates each Subscriber's partition.

If the Merge Agent detects that the function used in the filtering expression returns a value different from the one it returned for the initial snapshot, the merge or snapshot application fails, and you might need to reinitialize that Subscriber's subscription. However, it might be sufficient to change information at the Subscriber—for example, the login name—back to what its value was at the time of the original snapshot. The Merge Agent also checks whether the snapshot was generated prior to changes that invalidate it, such as metadata cleanup operations or schema changes. If a partitioned snapshot is too old, the Merge Agent will return an error and you need to regenerate a partitioned snapshot for that Subscriber based on a current regular snapshot.

Validating Subscriptions You can use SSMS, stored procedures, or replication management objects (RMOs) to validate all articles in a subscription. You need to use stored procedures to validate individual articles in snapshot and transactional publications.

To validate data at the Subscriber by using SSMS, you mark subscriptions for validation in the Validate Subscription, Validate Subscriptions, and Validate All Subscriptions dialog boxes, which are available from the Local Publications folder and the Local Subscriptions folder. You can also mark subscriptions from the All Subscriptions tab, the Subscription Watch List tab, and the Publications node in Replication Monitor.

A subscription is validated the next time it is synchronized by the Distribution Agent (for transactional replication) or the Merge Agent (for merge replication). The Distribution Agent typically runs continuously, in which case validation occurs immediately. The Merge Agent typically runs on demand, in which case validation occurs after you run the agent.

To view the validation results, you can access the detail windows in Replication Monitor, on the Distributor To Subscriber History tab for transactional replication or on the Synchronization History tab for merge replication. Alternatively, you can open the View Synchronization Status dialog box in SSMS. The validation results indicate whether validation succeeded or failed, but they do not specify which rows failed validation.

Push and Pull Subscriptions

A pull subscription is created and administered at the Subscriber. The Distribution Agent or Merge Agent for the subscription runs at the Subscriber and pulls information from the Publisher.

A push subscription is created and administered at the Publisher. The Distribution Agent or Merge Agent for the subscription runs at the Distributor and pushes information to the Subscriber.

Suppose, for example, a network of weather-testing stations use SQL Server 2005 to store data and to replicate this data to a central SQL server that analyzes it and creates reports. The readings need to be sent continuously to the central server. In this case, you create a transactional publication at each station and create push subscriptions to the central server.

Suppose, on the other hand, a central location held product information on a central server and sales outlets require updates periodically—for example, once per day. In this case, the sales outlets are the Subscribers and use pull subscriptions to obtain information from the central Publisher.

MORE INFO Pull subscriptions cannot be synchronized in Replication Monitor.

You can validate only push subscriptions by using Replication Monitor, because pull subscriptions cannot be synchronized in this tool. However, you can mark a subscription for validation and view validation results for pull subscriptions in Replication Monitor. For more information about push and pull subscriptions, search for “Subscribing to Publications” in Books Online or access [msdn2.microsoft.com/en-us/library/ms151170\(d=ide\).aspx](https://msdn2.microsoft.com/en-us/library/ms151170(d=ide).aspx).

You can also use replication stored procedures to validate that data at the Subscriber matches data at the Publisher. The procedures used depend on the replication type. The following Transact-SQL stored procedures are available for this purpose:

- *sp_publication_validation*
- *sp_article_validation*
- *sp_marksubscriptionvalidation*
- *sp_validatemergepublication*
- *sp_validatemergesubscription*

A third method of validating that data at the Subscriber matches data at the Publisher is to use RMOs to perform the validation programmatically. The objects you use depend on the replication type. If you use a program to monitor validation, you should configure the replication alert “Replication: Subscriber has failed data validation” to notify you when a failure occurs.

MORE INFO RMO programming

For more information, search for “How to: Synchronize a Push Subscription (RMO Programming),” “How to: Synchronize a Pull Subscription (RMO Programming),” and “How to: Programmatically Monitor Replication (RMO Programming)” in Books Online, or access [msdn2.microsoft.com/en-us/library/ms146910\(d=ide\).aspx](https://msdn2.microsoft.com/en-us/library/ms146910(d=ide).aspx), [msdn2.microsoft.com/en-us/library/ms147890\(d=ide\).aspx](https://msdn2.microsoft.com/en-us/library/ms147890(d=ide).aspx), and [msdn2.microsoft.com/en-us/library/ms147926\(d=ide\).aspx](https://msdn2.microsoft.com/en-us/library/ms147926(d=ide).aspx).

Exam Tip You need to be aware of the stored procedures that you can use to validate replication, and that RMOs give you the facility to write procedures that automatically validate replicated data at the Subscriber. However, examiners are unlikely to ask you to design such a procedure under examination conditions.

Data Validation Considerations You need to consider the following issues when validating data:

- You must stop all update activity at Subscribers before validating data. (You do not need to stop activity at the Publisher.)
- Because checksums and binary checksums can require large amounts of processor resources when you are validating a large data set, you should schedule validation to occur when there is the least activity on the servers that you use in replication.
- Replication validates tables only. It does not validate whether schema-only articles (for example, stored procedures) are the same at the Publisher and Subscriber.
- Binary checksum can validate any published table. Checksum cannot validate tables with column filters, or logical table structures where column offsets differ, for example, because of ALTER TABLE statements that drop or add columns. Checksum is deprecated in SQL Server 2005.
- Data in text, ntext, or image columns is not included in checksum calculations. The text, ntext, and image data types are deprecated in SQL Server 2005.
- Validation using binary checksum or checksum can erroneously report a failure when data types at the Subscriber are different from data types at the Publisher. This can happen, for example, if you initialize a subscription manually and are using different data types at the Subscriber.
- SQL Server 2005 does not support validation of data replicated to non-SQL Server Subscribers.
- While the calculations are performed, SQL Server 2005 places shared locks temporarily on tables for which row counts or checksums are being run. However, the calculations complete quickly and SQL Server 2005 removes the shared locks, usually in a matter of seconds.

Resolving Replication Conflicts

Merge replication allows multiple nodes to make autonomous data changes. In this case, situations exist in which a change made at one node might conflict with a change made to the same data at another node. Sometimes the Merge Agent can encounter an error—for example, a constraint violation—and as a result cannot propagate a change made at a particular node to another node. This section describes the types of conflicts that can occur, how such conflicts are detected and resolved, and the factors that affect detection and resolution.

Detecting and Resolving Conflicts

The Merge Agent uses the lineage column of the *MSmerge_contents* system table to detect conflicts. If column-level tracking is enabled for an article, it also uses the COLV1 column. These columns contain metadata about when a row or column is inserted or updated, and about the nodes in a merge replication topology that made changes to the row or column. You can use the Transact-SQL system stored procedure *sp_showrowreplicainfo* to view this metadata.

The Merge Agent compares the metadata for each row at the Publisher and Subscriber, and it uses this metadata to determine whether a row or column has changed at more than one node in the topology, thus indicating a potential conflict. If a conflict is detected, the Merge Agent launches the *conflict resolver* specified for the article with a conflict, and it uses the resolver to determine the conflict winner. The winning row is applied at the Publisher and Subscriber, and the data from the losing row is written to a conflict table.

The Merge Agent resolves conflicts automatically unless you choose interactive conflict resolution for the article. If you manually change the winning row for a conflict using the merge replication Conflict Viewer, the Merge Agent applies the winning version of the row to the losing server during the next synchronization.

MORE INFO Interactive conflict resolution

For more information about Interactive Resolver, which allows you to resolve conflicts manually during on-demand synchronization, search for "Interactive Conflict Resolution" in Books Online or access [msdn2.microsoft.com/en-us/library/ms151317\(d=ide\).aspx](https://msdn2.microsoft.com/en-us/library/ms151317(d=ide).aspx).

Logging Resolved Conflicts

After the Merge Agent has resolved the conflict according to the logic in the conflict resolver, it logs conflict data according to the type of conflict. For UPDATE and INSERT conflicts, it writes the losing version of the row to the conflict table for the article. General conflict information—for example, the type of conflict—is written to the *MSmerge_conflicts_info* table. For DELETE conflicts, it writes the losing version of the row to the *MSmerge_conflicts_info* table. When a delete loses against an update, no data exists for the losing row (because it was a delete), so nothing is written to the conflict table.

The Merge Agent creates conflict tables for each article in the publication database, the subscription database, or (by default) both, depending on the value specified

for the *@conflict_logging* parameter of *sp_addmergepublication*. Each conflict table has the same structure as the article on which it is based, with the addition of the *origin_datasource_id* column. The Merge Agent deletes data from the conflict table if the data is older than the conflict retention period for the publication, which you can specify by using the *@conflict_retention* parameter of *sp_addmergepublication*. (The default is 14 days.) SQL Server 2005 replication provides the Replication Conflict Viewer and the stored procedures *sp_helpmergearticleconflicts*, *sp_helpmergeconflictrows*, and *sp_helpmergedeleteconflictrows* to view conflict data.

Subscription Types and Conflict Tracking

The following factors affect how the Merge Agent resolves a conflict it detects:

- **Type of subscription** Client or server (whether the subscription is a pull subscription or a push subscription does not affect conflict resolution)
- **Type of conflict tracking** Row-level, column-level, or logical record-level

Subscription Types When you create a subscription, you specify whether it is a push or pull subscription and whether it is a client or server subscription. After a subscription is created, the type cannot be changed. (In previous versions of SQL Server, client and server subscriptions were, respectively, local and global subscriptions.)

If you assign a fixed priority value (from 0.00 through 99.99), you create a server subscription. If you initially set a priority value of 0.00 but configure the subscription to assume and retain the priority value of the Publisher after synchronization, you create a client subscription. Subscribers with server subscriptions can republish data to other Subscribers. You can use client subscriptions when you want all Subscribers to have the same priority, and the first Subscriber to merge with the Publisher to win the conflict.

When you change a row in a server subscription, SQL Server 2005 stores the subscription priority in the metadata for the change. This priority value travels with the changed row as it merges with changes at other Subscribers, ensuring that a change made by a higher priority subscription does not lose to a subsequent change made by a subscription with a lower priority.

Delayed Conflict Notification

Delayed conflict notification can occur with server subscriptions that have different conflict priorities—for example, when changes are exchanged between the Publisher and a lower priority Subscriber that result in conflicting changes when

a higher priority Subscriber later synchronizes with the Publisher. This situation can become problematic when the lower priority Subscriber has made changes to the same rows that are now conflict losers, resulting in a loss of all the changes made by this Subscriber. A potential solution to this problem is to ensure that all the Subscribers have the same priority, unless business logic dictates otherwise.

Tracking Level Whether a data change qualifies as a conflict depends on the type of conflict tracking you set for an article: row-level, column-level, or logical record-level. When conflicts are detected at the row level, SQL Server 2005 assumes that changes made to corresponding rows are in conflict, regardless of whether the changes are made to the same column. For example, suppose a change is made to the address column of a Publisher row, and a second change is made to the name column of the corresponding Subscriber row in the same table. Row-level tracking detects a conflict because changes were made to the same row. Column-level tracking does not detect a conflict because changes were made to different columns in the same row. If a conflict is detected by either row-level or column-level tracking, the entire row of data is overwritten by data from the conflict winner.

SQL Server 2005 introduces logical record-level tracking. The method of detecting conflicts for logical records is determined by two article properties: *column_tracking* and *logical_record_level_conflict_detection*. You can set the *logical_record_level_conflict_detection* article property (for the top-level parent article only) to TRUE or FALSE. If this value is FALSE, merge replication detects conflicts based solely on the value of the *column_tracking* property for the article (as in previous versions of SQL Server). If the value is TRUE, merge replication will ignore the *column_tracking* property of the article and detect a conflict if changes are made anywhere in the logical record.

If you are updating customer data that typically changes at several locations at the same time, you should choose row-level tracking. If you were to choose column-level tracking in this situation, SQL Server 2005 would not detect changes to the customer address in one location and to the customer name in another location as a conflict, the data would be merged on synchronization, and the error would be missed.

Suppose, on the other hand, that your SQL Server enterprise contains two sites that have access to different types of statistical information—for example, income level and total value of credit card purchases. In this case, you select column-level tracking to ensure that both sites can update statistical data for different columns without generating unnecessary conflicts.

BEST PRACTICES Choose row-level tracking whenever possible.

If your application does not require column-level tracking, Microsoft recommends that you use row-level tracking (the default) because it typically results in better synchronization performance.

Conflict Types

Although updates cause the majority of conflicts, other conflict types exist. Table 10-4 lists the various conflict types and the corresponding value for each type in the `conflict_type` column in the `MSmerge_conflicts_info` table.

Table 10-4 Conflict Types

Description	Value in conflict_type Column
Update Conflict	1
Column Update Conflict	2
Update Delete Wins Conflict	3
Update Wins Delete Conflict	4
Upload Insert Failed	5
Download Insert Failed	6
Upload Delete Failed	7
Download Delete Failed	8
Upload Update Failed	9
Download Update Failed	10
Resolution	11
Logical Record Update Wins Delete Conflict	12
Logical Record Conflict Insert Update	13
Logical Record Delete Wins Update Conflict	14

The conflict types listed in Table 10-4 can be summarized as follows:

- Update-Update Conflicts** The Merge Agent detects an update-update conflict when an update to a row, column, or logical record at one node conflicts with a

second update to the same row at another node. The behavior of the default resolver in this case is to send the winning version of the row to the losing node and log the losing row version in the article conflict table.

- **Update-Delete Conflicts** The Merge Agent detects an update-delete conflict when an update of data at one node conflicts with a delete at another. In this case, the Merge Agent updates a row, but when it searches for that row at the destination it has been deleted. If the winner is the node that updated the row, the delete at the losing node is discarded and the Merge Agent sends the newly updated row to the conflict loser. The Merge Agent logs information about the losing version of the row to the *MSmerge_conflicts_info* table.
- **Failed Change Conflicts** The Merge Agent detects a failed change conflict when it cannot apply a particular change. This typically occurs because of a difference in constraint definitions between the Publisher and Subscriber, combined with the setting in the NFR option on the constraint. Examples include the following:
 - A foreign key conflict at the Subscriber, which can occur when the Subscriber-side constraint is not marked as NFR.
 - Differences in constraints between the Publisher and Subscribers when the constraints are not marked as NFR.
 - Unavailability of dependent objects at the Subscriber. For example, if you publish a view but not the table on which that view depends, a failure occurs when you attempt to insert that view at the Subscriber.
 - Join filtering logic for a publication that does not match the primary key and foreign key constraints. Conflicts can occur when the SQL Server relational engine tries to honor a constraint but the Merge Agent is honoring the join filter definition between the articles. The Merge Agent cannot apply the change at the destination node because of the table-level constraints, which results in a conflict.
 - Conflicts because of unique index or unique constraint violations or primary key violations. These can occur if identity columns are defined for the article and automated identity range management is not used. This is a problem when two Subscribers use the same identity value for a newly inserted row.

MORE INFO Automated identity range management

For more information about automated identity range management, search for "Replicating Identity Columns" in Books Online or access [msdn2.microsoft.com/en-us/library/ms152543\(d=ide\).aspx](https://msdn2.microsoft.com/en-us/library/ms152543(d=ide).aspx).

- ❑ Conflicts because of trigger logic preventing the Merge Agent from inserting a row in the destination table—for example, an update trigger defined at the Subscriber is not marked as NFR and includes a ROLLBACK in its logic. When a failure occurs, the trigger issues a ROLLBACK of the transaction, resulting in the Merge Agent detecting a failed change conflict.

Configuring Agent Profiles

Configuring SQL Server 2005 replication results in the installation of a set of agent profiles on the Distributor. An agent profile contains a set of parameters that the agent uses each time it runs. Each agent logs in to the Distributor during its startup process and queries for the parameters in its profile. If the profile changes, the Merge Agent updates the profile at the Subscriber the next time it runs.

NOTE Merge subscriptions that use Web synchronization

For merge subscriptions that use Web synchronization, SQL Server 2005 downloads and stores profiles at the Subscriber.

SQL Server 2005 provides the following Replication Agents as executable files:

- **Snapshot Agent** Prepares snapshot files containing the schema and data of published tables and database objects, stores the files in the snapshot folder, and records synchronization jobs in the distribution database.
- **Log Reader Agent** Monitors the transaction log of each database configured for transactional replication, and copies the transactions marked for replication from the transaction log into the distribution database.
- **Distribution Agent** Moves the snapshot (for snapshot replication and transactional replication) and the transactions held in the distribution database tables (for transactional replication) to the destination tables at the Subscribers.
- **Merge Agent** Applies the initial snapshot held in the database tables to the Subscribers. It also merges incremental data changes that occurred at the Publisher after the initial snapshot was created, and it reconciles conflicts either according to the rules you configure or by using a custom resolver you create.
- **Queue Reader Agent** Reads messages stored in a SQL Server queue or a Message Queue, and then applies those messages to the Publisher. Queue Reader Agent is used with snapshot and transactional publications that allow queued updating.

SQL Server 2005 replication provides a default profile for each agent and additional predefined profiles for the Log Reader Agent, Distribution Agent, and Merge Agent. You can also create profiles suited to your application requirements. Only one profile is active for an agent at any one time.

An agent profile allows you to change key parameters for all agents associated with that profile. For example, if you have 20 Snapshot Agents and need to change the query timeout value (the *-QueryTimeout* parameter), you can update the profile used by the Snapshot Agents. All Snapshot Agents then automatically use the new query timeout value the next time they run.

MORE INFO Replication agent profile parameters

For more information about the parameters specified in the various replication agent profiles, search for "Replication Agent Profiles" in Books Online or access [msdn2.microsoft.com/en-us/library/ms151223\(d=ide\).aspx](http://msdn2.microsoft.com/en-us/library/ms151223(d=ide).aspx).

You can configure different profiles for different instances of an agent. For example, a Merge Agent that connects to the Publisher and Distributor over a slow satellite connection could use a set of parameters that are better suited to the slower communications link by using the *slow link* profile.

CAUTION Command-line parameters override profile settings.

If you specify a value for an agent parameter on the command line, that value overrides the value set for the same parameter in the agent profile.

Managing Agent Profiles

You can use Replication Monitor and SSMS to view, configure, create, and delete agent profiles. Transact-SQL stored procedures also enable you to create new agent profiles. Replication Monitor provides the following dialog boxes for managing agent profiles:

- **Agent Profiles** Enables you to change the properties of profiles, create and delete profiles, specify a default profile, and specify that all agents of a specific type (for example, snapshot agents) should use a given profile. To create a new profile, you access the New Agent Profile dialog box from this dialog box. Do not modify or delete predefined profiles.
- **Properties** Enables you to view and edit the parameter settings in a specified profile.

- **New Agent Profile** Enables you to create a new profile, and optionally include the values from an existing profile.

IMPORTANT Stop and restart agents that run continuously.

Agent parameter changes take effect the next time the agent is started. If the agent runs continuously, you must stop and restart the agent before the profile changes will take effect.

SSMS provides an alternative method of accessing the Agent Profiles dialog box. In SSMS, you click Profile Default on the General page of the Distributor Properties - <Distributor> dialog box.

MORE INFO Working with agent profiles

For detailed instructions about how to create, delete, and configure replication agent profiles, search for "How to: Work with Replication Agent Profiles (SQL Server Management Studio)" in Books Online or access [msdn2.microsoft.com/en-us/library/ms152515\(d=ide\).aspx](http://msdn2.microsoft.com/en-us/library/ms152515(d=ide).aspx).

You can use Transact-SQL stored procedures to create, modify, and remove agent profiles and specify that an agent should use a profile during synchronization. SQL Server 2005 provides the following stored procedures for these purposes:

- ***sp_add_agent_profile*** Creates a new agent profile
 - ***sp_help_agent_profile*** Modifies or removes an existing agent profile, and specifies that an agent uses a profile during synchronization
-

MORE INFO Using stored procedures

For more information about how to use Transact-SQL stored procedures to manage replication agent profiles, search for "How to: Work with Replication Agent Profiles (Replication Transact-SQL Programming)" in Books Online or access [msdn2.microsoft.com/en-us/library/ms147893\(d=ide\).aspx](http://msdn2.microsoft.com/en-us/library/ms147893(d=ide).aspx).

Tuning Replication Configuration

You can enhance the general performance for all types of SQL Server replication by tuning your server and network performance, following best practices for database design, optimizing your Publisher so that you publish only the information you need to publish, tuning your subscriptions, optimizing snapshots (used in all types of replication), and configuring your replication agent parameters.

Tuning Your Server and Network

You can tune your server to maximize replication efficiency by specifying the minimum and maximum amounts of memory allocated to the SQL Server 2005 database engine. By default, the database engine changes its memory requirements dynamically based on available system resources. You can use the Min Server Memory option to set the minimum available memory and avoid low memory availability during replication activities.

To avoid having the operating system page to hard disk, you can also set a maximum amount of memory with the Max Server Memory option. The precise levels of these settings will depend on the role of the server in replication and what other tasks it carries out. The settings typically vary from server to server within your enterprise. No single *optimal* setting exists.

MORE INFO Server memory options

For more information, search for “Server Memory Options” in Books Online or access [msdn2.microsoft.com/en-us/library/ms178067\(d=ide\).aspx](http://msdn2.microsoft.com/en-us/library/ms178067(d=ide).aspx).

You should also consider increasing the amount of random access memory (RAM) on servers involved in replication, especially on the server (or servers) that contains the Distributor, and specifying multiprocessor machines for your replication servers. Replication agents can take advantage of additional processors on the server.

You can improve performance, sometimes dramatically, by ensuring you have efficiently allocated disk storage for database files and log files. Whether you have configured replication or not, you can improve performance and disaster recovery by storing transaction logs on a separate disk or disk array from database files. If you configure replication, you should also consider separate disks for all databases involved in the replication process. Chapter 1, “Troubleshooting Database and Server Performance,” and Chapter 4, “Disaster Recovery,” discuss disk array solutions.

The network can be a significant performance bottleneck, particularly for transactional replication. You can significantly enhance the propagation of changes to Subscribers by using 100-megabit or gigabit ethernet. If your network is slow, you need to specify appropriate network settings and agent parameters.

MORE INFO Slow network settings

For more information about the appropriate settings for slow networks, search for “A Slow Network Is Causing Problems” in Books Online or access [msdn2.microsoft.com/en-us/library/ms151858\(d=ide\).aspx](http://msdn2.microsoft.com/en-us/library/ms151858(d=ide).aspx).

Tuning Database Design

A replicated database generally benefits from the same performance optimization as a nonreplicated database. Chapter 1 and Chapter 2, “Analyzing Queries,” discuss database and query performance optimization. However, you need to use indexes cautiously, especially at the Subscriber. You should index the primary key column at the Subscriber, but additional indexes can adversely affect insert, update, and delete performance.

You can reduce contention between user activity and replication agent activity by setting the `READ_COMMITTED_SNAPSHOT` database option to `ON` for the publication and subscription databases. When you enable this option, transactions specifying the `READ_COMMITTED` isolation level (the default) use row versioning instead of locking. When a transaction runs all statements, you can see a snapshot of data as it exists at the start of the statement.

You need to take care when using triggers. Business logic in user-defined triggers at a Subscriber can slow down the replication of changes to that Subscriber. For transactional replication, you can increase efficiency by including this logic in custom stored procedures used to apply the replicated commands. For merge replication, you can increase efficiency by using business logic handlers. If you use triggers to maintain referential integrity in tables published for merge replication, you should specify the processing order of tables to reduce the number of retries required for the Merge Agent.

MORE INFO Triggers and business logic

For more information about how to handle triggers and business logic in transactional and merge replication, search for “Specifying How Changes Are Propagated for Transactional Articles,” “Executing Business Logic During Merge Synchronization,” and “Specifying the Processing Order of Merge Articles” in Books Online, or access [msdn2.microsoft.com/en-us/library/ms152489\(d=ide\).aspx](http://msdn2.microsoft.com/en-us/library/ms152489(d=ide).aspx), [msdn2.microsoft.com/en-us/library/ms152495\(d=ide\).aspx](http://msdn2.microsoft.com/en-us/library/ms152495(d=ide).aspx), and [msdn2.microsoft.com/en-us/library/ms152469\(d=ide\).aspx](http://msdn2.microsoft.com/en-us/library/ms152469(d=ide).aspx).

Large Objects (LOBs) require more storage space and processing than other column data types. You should not include LOBs in articles unless they are necessary for your application. If you do include LOBs, Microsoft recommends that you use the `varchar(max)`, `nvarchar(max)`, and `varbinary(max)` data types. The `text`, `ntext`, and `image` data types are deprecated in SQL Server 2005.

Tuning Publication Design

You can minimize conflicts by tuning publication design. Conflicts can occur when your replication configuration permits data updates at Subscribers. Merge replication, transactional replication with updatable subscriptions, and peer-to-peer transactional

replication permit this functionality. Merge replication and transactional replication with updatable subscriptions support data conflicts if a given row is updated at more than one node between synchronizations. Peer-to-peer replication does not support data conflicts, and data changes must be partitioned. However, regardless of the type of replication you use, you should partition changes whenever possible. This reduces the processing required for conflict detection and resolution.

You can partition changes by publishing subsets of data to each Subscriber or by using an application that directs changes for a given row to a given node. Merge replication supports publishing subsets of data using parameterized filters with a single publication. Transactional replication supports publishing subsets of data using static filters with multiple publications.

When a transactional publication includes one or more articles that use row filters, the Log Reader Agent must apply the filter to each row affected by an update to the table as it scans the transaction log. This reduces the throughput of the Log Reader Agent. Merge replication evaluates changed or deleted rows to determine which Subscribers receive those rows. If you use row filters to reduce the data required at a Subscriber, this processing is more complex and can be slower than when you publish all rows in a table. You need to carefully balance the tradeoff between reduced storage requirements at each Subscriber and the need to achieve maximum throughput.

MORE INFO Row filters

For more information about filtering, search for "Filtering Published Data" in Books Online or access [msdn2.microsoft.com/en-us/library/ms151775\(d=ide\).aspx](https://msdn2.microsoft.com/en-us/library/ms151775(d=ide).aspx).

Real World

Ian McLean

I've found that the best method of increasing the efficiency of Publishers is also the simplest. Cut down the amount of information that you publish. Because replication is fairly straightforward to configure, and because there's a remote possibility that a table might be useful to staff at a distant location, most DBAs publish more data than is required and replicate it more often than they need to. This uses additional resources within the distribution databases and snapshot files, and lowers the throughput for required data. Even more significantly, if you provide more data than users require, it takes those users longer to find the information they need. Do not publish an entire database when all the information remote users will ever need is in a single table.

Tuning Subscriptions

The Distribution Agent and Merge Agent run at the Distributor for push subscriptions and at Subscribers for pull subscriptions. Using pull subscriptions moves agent processing from the Distributor to the Subscribers. This approach can improve performance significantly, particularly if you have a large number of Subscribers.

If you need to send a large number of changes to Subscribers, it might be faster and more efficient to reinitialize them with a new snapshot rather than using replication. To help you make this judgment, you can access the Undistributed Commands tab in Replication Monitor. This tab displays (for transactional replication) the number of transactions in the distribution database that have not yet been distributed to a Subscriber and the estimated time for distributing these transactions.

Tuning Snapshots

The Snapshot Agent bulk copies data from the published table on the Publisher to a file in the snapshot folder on the Distributor. You should schedule this resource-intensive process to occur during off-peak times. Unless you are sending snapshots to non-SQL Server Subscribers or Subscribers running SQL Server Mobile Edition, which require character mode snapshots, you should use native mode snapshots (the default). Native mode snapshots use fewer resources.

You can choose to generate snapshot files in the default snapshot folder, in an alternate snapshot folder, or in both. Generating snapshot files at both locations requires additional disk space and more processing when the Snapshot Agent runs. Unless other considerations dictate otherwise, you should place the snapshot folder on a drive local to the Distributor that is not used to store database or log files. This approach reduces contention among the disks and helps the snapshot process complete faster.

When you create the subscription database at the Subscriber, specify a simple or bulk-logged recovery model. Doing this allows minimal logging of the bulk inserts performed during the application of the snapshot at the Subscriber. If necessary, you can change to a different recovery model after the snapshot has been applied to the subscription database.

Consider carefully whether you should compress your snapshot files. Compressed snapshots can improve performance when transferring snapshot files across a network. However, compressing the snapshot requires additional processing by the Snapshot Agent when it generates the snapshot files, and by the Distribution Agent or

Merge Agent when applying the snapshot files. Additionally, compressed snapshots cannot be resumed if a network failure occurs and are therefore not suitable for unreliable networks.

MORE INFO **Initializing a subscription manually**

If a subscription involves a large initial dataset, initializing the subscription manually can sometimes be more efficient than using a snapshot. For more information, search for "Initializing a Transactional Subscription Without a Snapshot" and "Initializing a Merge Subscription Without a Snapshot" in Books Online or access [msdn2.microsoft.com/en-us/library/ms151705\(d=ide\).aspx](http://msdn2.microsoft.com/en-us/library/ms151705(d=ide).aspx) and [msdn2.microsoft.com/en-us/library/ms152488\(d=ide\).aspx](http://msdn2.microsoft.com/en-us/library/ms152488(d=ide).aspx).

Tuning Agent Parameters

Except during initial testing, monitoring, or debugging, you should reduce the *-HistoryVerboseLevel* parameter and the *-OutputVerboseLevel* parameter of the Distribution Agents or Merge Agents. Doing this reduces the number of new rows inserted to track agent history and output.

You should use the *-MaxBCPThreads* parameter of the Snapshot Agent, Merge Agent, and Distribution Agent to ensure that the number of threads specified does not exceed the number of processors on the computer. This parameter specifies the number of bulk copy operations that the database engine performs in parallel when it creates and applies the snapshot.

Unless your published tables include extensible markup language (XML) columns, you should use the *-UseInprocLoader* parameter of the Distribution Agent and the Merge Agent to cause the agent to use the BULK INSERT command when the snapshot is applied.

Using Replication with Database Mirroring

SQL Server 2005 introduces database mirroring, which you can use in conjunction with replication to provide availability for the publication database. Database mirroring involves two copies of a single database that typically reside on different computers. At any given time, only one copy of the database, known as the *principal database*, is available to clients. Updates made by clients to the principal database are applied on the other copy of the database, known as the *mirror database*. Mirroring involves applying the transaction log from every insertion, update, or deletion made on the principal database onto the mirror database.

IMPORTANT Database mirroring and SP1

Prior to the release of SP1, Microsoft support policies did not apply to the database mirroring feature in SQL Server 2005. Database mirroring was disabled by default, and could be enabled for evaluation purposes only by using trace flag 1400 as a startup parameter. However, in SQL Server 2005 SP1, database mirroring is enabled by default and Microsoft supports its use with replication on production databases. Trace flag 1400 is deprecated. A new Database Mirroring monitor has been added to SSMS. To start this monitor, you right-click a database node, choose Tasks, and then choose Launch Database Mirroring Monitor.

Replication failover to a mirror is supported for publication databases only, and it is not supported for the distribution database or subscription databases.

NOTE Mirror and principal databases

After a failover, the mirror becomes the principal. In this section, “principal” and “mirror” refer to the original principal and mirror databases.

You need to be aware of the following requirements and considerations when using replication with database mirroring:

- The principal and mirror must share a Distributor. Microsoft recommends that you use a remote Distributor because this provides greater fault tolerance if the Publisher has an unplanned failover.
- The Publisher and Distributor must be SQL Server 2005 servers. Subscribers can run SQL Server 2005 or a previous version, but functionality is limited if Subscribers are not SQL Server 2005 servers.
- Replication supports mirroring the publication database for merge or transactional replication with read-only Subscribers or queued updating Subscribers. Immediate updating Subscribers, Oracle Publishers, Publishers in a peer-to-peer topology, and republishing are not supported.
- Metadata and objects that exist outside the database are not copied to the mirror. These include logins, jobs, and linked servers. If you require the metadata and objects at the mirror, you need to copy them manually.

Configuring Replication with Database Mirroring

The top-level procedure for configuring replication with database mirroring consists of the following steps:

1. Configure the Publisher.
2. Configure database mirroring.

3. Configure the mirror to use the same Distributor as the principal.
4. Configure replication agents for failover.
5. Add the principal and mirror to Replication Monitor.

MORE INFO Detailed procedure

For more information and detailed instructions on configuring replication with database mirroring, search for “Replication and Database Mirroring” in Books Online or access [msdn2.microsoft.com/en-us/library/ms151799\(d=ide\).aspx](http://msdn2.microsoft.com/en-us/library/ms151799(d=ide).aspx).

Maintaining a Mirrored Publication Database

Maintaining a mirrored publication database is similar to maintaining a non-mirrored database, with the following additional considerations:

- Administration and monitoring must occur at the active server. In SSMS, publications appear under the Local Publications folder only for the active server. For example, if you fail over to the mirror, the publications are displayed at the mirror and are no longer displayed at the principal. If the database fails over to the mirror, you might need to manually refresh SSMS and Replication Monitor to reflect the change.
- Replication Monitor displays Publisher nodes in the object tree for both the principal and the mirror. If the principal is the active server, publication information is displayed only under the principal node in Replication Monitor.
- If the mirror is the active server and if an agent has an error, the error is indicated only on the principal node, not on the mirror node. If the principal is unavailable, the principal and mirror nodes display identical lists of publications. Monitoring should be performed on the publications under the mirror node.
- When you use stored procedures or RMOs to administer replication at the mirror and you need to specify the Publisher name, you must specify the name of the instance on which the database was enabled for replication. You can use the Transact-SQL function `PUBLISHINGSERVERNAME` to determine the appropriate name.

When a publication database is mirrored, the replication metadata stored in the mirrored database is identical to the metadata stored in the principal database. Consequently, for publication databases enabled for replication at the principal, the Publisher instance name stored in system tables at the mirror is the name of the principal, not the name of the mirror. This arrangement affects replication configuration and maintenance if the publication database fails over to the mirror. For example, if

you are configuring replication with stored procedures on the mirror after a failover, and you want to add a pull subscription to a publication database that was enabled at the principal, you must specify the principal name rather than the mirror name for the *@publisher* parameter of *sp_addpullsubscription* or *sp_addmergepullsubscription*.

If you enable a publication database at the mirror after failover to the mirror, the Publisher instance name stored in system tables is the name of the mirror. In this case, you would use the name of the mirror for the *@publisher* parameter. To synchronize a subscription in Management Studio after a failover, you need to synchronize pull subscriptions from the Subscriber and synchronize push subscriptions from the active Publisher.

Replication Behavior if Mirroring Is Removed

You need to keep the following issues in mind if database mirroring is removed from a published database:

- If the publication database at the principal is no longer mirrored, replication continues to work unchanged against the original principal.
- If the publication database fails over from the principal to the mirror and the mirroring relationship is subsequently disabled or removed, replication agents will not function against the mirror. If the principal is permanently lost, you need to disable and then reconfigure replication with the mirror specified as the Publisher.
- If database mirroring is removed completely, the mirror database is in a recovery state and must be restored to become functional. The behavior of the recovered database with respect to replication depends on whether the `KEEP_REPLICATION` option is specified. This option forces the restore operation to preserve replication settings when restoring a published database to a server other than that on which the backup was created.

Exam Tip For the purposes of the exam, you should assume that database mirroring is a feature of SQL Server 2005. Microsoft examiners typically pose questions based on significant new features of a product, and the installation of the current service pack is assumed, unless the question specifically states otherwise.

Log Shipping

Microsoft recommends that you use database mirroring rather than log shipping. However, log shipping remains a valid technique for disaster recovery and you need to be aware of it.

Log shipping involves two copies of a single database that typically reside on different computers. At any given time, only one copy of the database is available to clients. This copy is known as the *primary database*. Log shipping involves backing up the transaction log for the primary database and running the backup against the secondary database.

You can use log shipping in conjunction with replication. Replication does not continue after a log shipping failover. If a failover occurs, replication agents do not connect to the secondary and transactions are not replicated to Subscribers. Replication resumes if a failback to the primary occurs. All transactions that log shipping copies from the secondary back to the primary are replicated to Subscribers. If the primary is permanently lost, the secondary can be renamed to enable replication to continue.

PRACTICE Configuring and Verifying Replication

In this practice session, you configure the Central Publisher model for transactional replication, with your member server as the Publisher and Distributor and your domain controller as the Subscriber. You create a publication that contains the HumanResources.Employees table of the AdventureWorks database on your member server, create a push subscription to the same database on your domain controller, and then verify replication and measure replication latency. You need to complete Practice 1 before you start Practice 2. You need to complete Practices 1 and 2 before you start Practice 3. If you want to expand this practice session, use the ALTER TABLE Transact-SQL statement to make changes to the HumanResources.Employee table in the AdventureWorks database on your member server and check that these changes replicate to the same table on your domain controller.

► Practice 1: Configuring Transactional Replication

In this practice, you configure transactional replication, specify the Central Publisher model, and create a transactional publication on your member server.

1. Log in to your domain at your member server by using either your domain administrator account or an account that has been added to the sysadmin server role. If you are using virtual machine software, log in to your domain and connect to your member server.
2. Create and share a folder to hold your snapshots—for example, C:\Snapshot. Accept the default share and NTFS permissions.

3. From Programs, choose Microsoft SQL Server 2005, and then choose SQL Server Management Studio. Connect to the database engine on your member server, specifying Windows Authentication. Connect using TCP/IP, and specify the AdventureWorks database.
4. In the left-hand pane, expand your server.
5. Expand the Replication folder, and then right-click the Local Publications folder.
6. Choose New Publication. The New Publication Wizard opens as shown in Figure 10-5. Click Next.



Figure 10-5 The New Publication Wizard.

7. On the Distributor page, specify that the Publisher server will act as its own Distributor (a local Distributor—the default). Click Next. The New Publication Wizard automatically configures the server.
8. On the Snapshot Folder page, specify the default snapshot folder that you created earlier for the Distributor, as shown in Figure 10-6. (If you want to configure pull subscriptions, you need to enter a network path—for example, \\GLASGOW\\Snapshot.) Click Next.

NOTE Specifying a remote Distributor

If you specify that another server should act as the Distributor, you must enter a password on the Administrative Password page for connections made from the Publisher to the Distributor. This password must match the password specified when the Publisher was enabled at the remote Distributor.

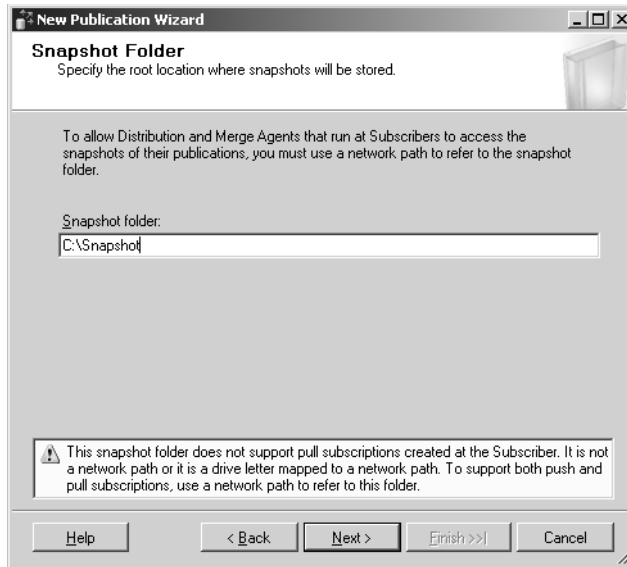


Figure 10-6 Specifying the snapshot folder.

9. Choose a publication database—in this case, AdventureWorks. Click Next.
10. Select Transactional Publication as shown in Figure 10-7. Click Next.

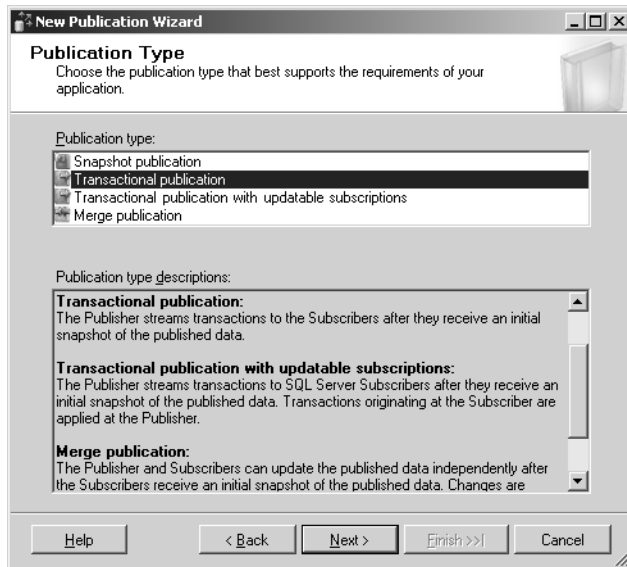


Figure 10-7 Specifying the publication type.

11. In the Objects To Publish pane of the Articles page, expand Tables and select the check box for the HumanResources.Employee table as shown in Figure 10-8. Click Next.

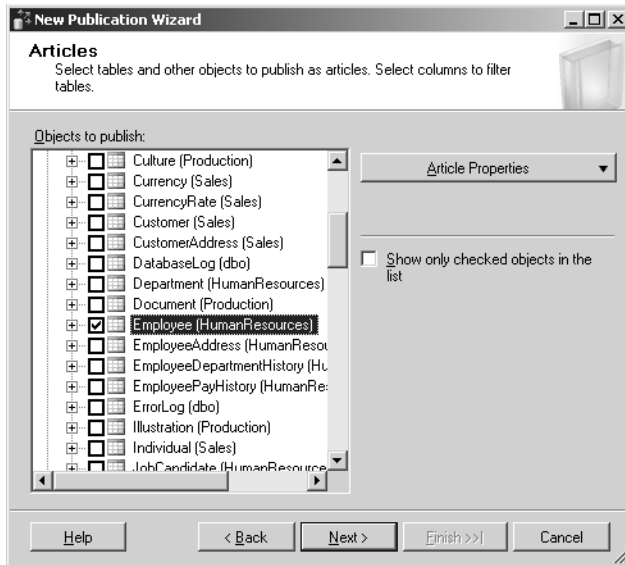


Figure 10-8 Specifying the articles to publish.

12. In this practice, you do not need to filter any rows. Click Next.
13. Select the Create A Snapshot Immediately And Keep The Snapshot Available To Initialize Subscriptions check box, as shown in Figure 10-9. Click Next.

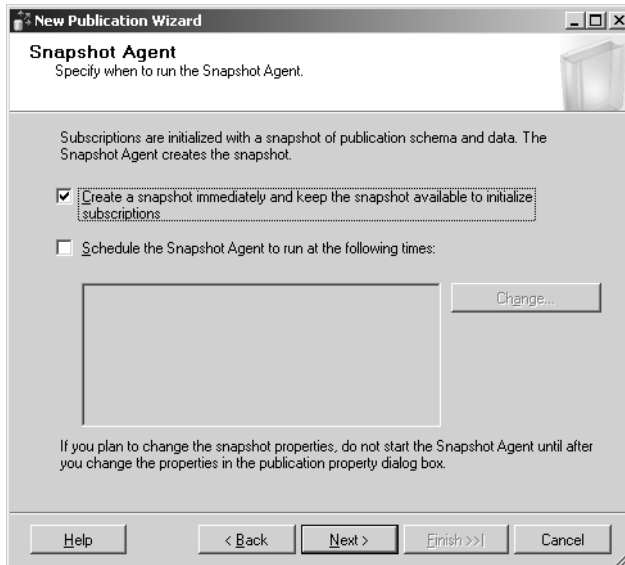


Figure 10-9 Specifying when the Snapshot Agent runs.

- On the Agent Security page, ensure that the Use The Security Settings From The Snapshot Agent check box is selected as shown in Figure 10-10. Click Security Settings for the Snapshot Agent.

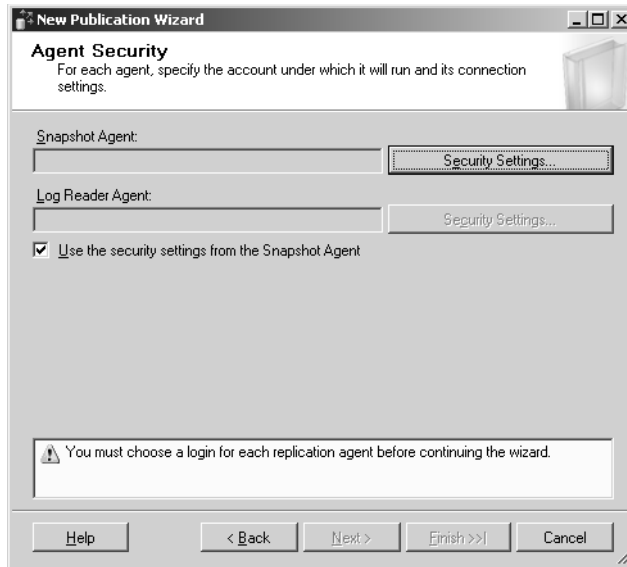


Figure 10-10 Using the security settings from the Snapshot Agent.

- Specify the credentials under which the Snapshot Agent process will run. Specify that replication connects to the Publisher by impersonating the process account, as shown in Figure 10-11. Click OK. Click Next.

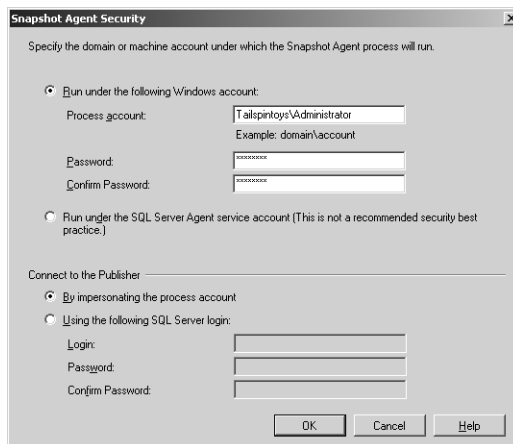


Figure 10-11 Specifying the credentials under which the Snapshot Agent will run.

IMPORTANT Using the Administrator account

In a production environment, you would rename the Administrator account and create a second account with the appropriate administrative privileges. You would not use an account called Administrator. The practices in this book use the Administrator account both for convenience and to indicate the privilege level required to carry out the procedures.

16. On the Wizards Actions page, ensure that the Create The Publication check box is selected. Click Next.
17. Give the publication a name—for example, MyPub. Click Finish. The New Publication Wizard creates the publication as shown in Figure 10-12. Click Close.

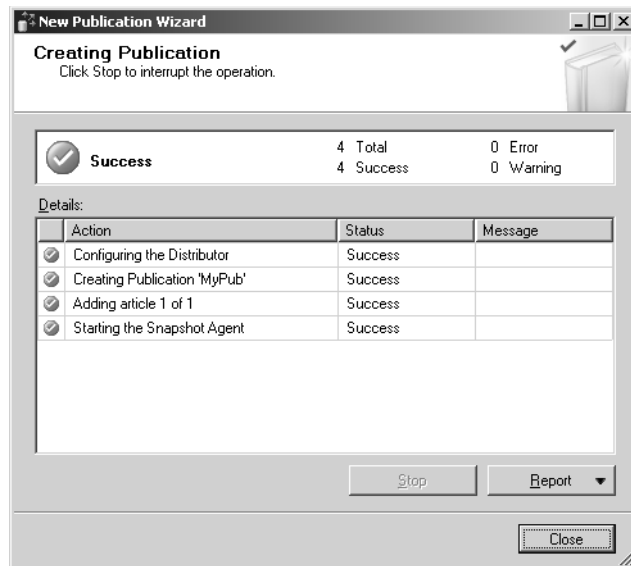


Figure 10-12 Creating the publication.

► **Practice 2: Creating a Push Subscription from the Publisher**

In this practice, you create a push subscription from the Publisher to a subscription database on the Subscriber. (You need to complete Practice 1 before you attempt Practice 2.) If you are starting Practice 2 directly after Practice 1 and SSMS is still open, proceed directly to the first step in this practice. If not, log in, start SSMS, and connect to your member server as described in Practice 1.

1. If necessary, expand the server node in SSMS, expand the Replication folder, and then expand the Local Publications folder.

2. Right-click [AdventureWorks]: MyPub and then choose New Subscriptions.
3. The New Subscription Wizard starts as shown in Figure 10-13. Click Next.

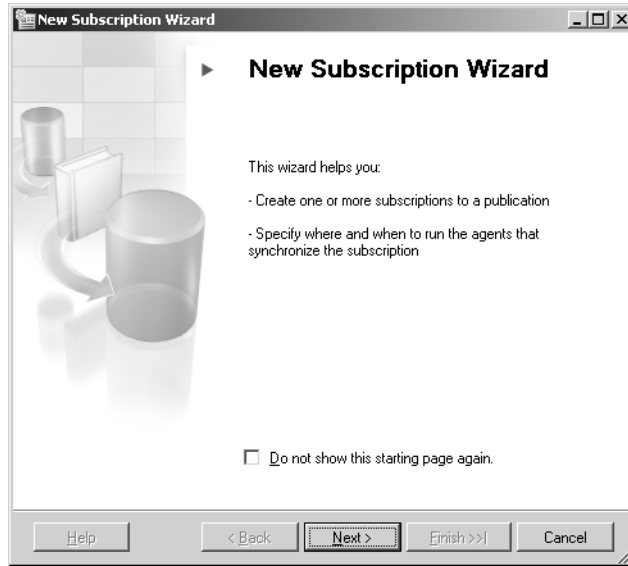


Figure 10-13 The New Subscription Wizard.

4. On the Publication page, ensure that MyPub is selected. Click Next.
5. On the Distribution Agent Page, ensure Run All Agents At The Distributor, GLASGOW [Push Subscriptions] is selected. Click Next.
6. On the Subscribers page, click Add Subscriber. Select Add SQL Server Subscriber as shown in Figure 10-14. (If you have set up your test network as recommended, GLASGOW appears in the Subscriber list rather than OFFICE.)
7. In the Server Name text box in the Connect To Server dialog box, specify your domain controller (MELBOURNE). On the Options tab, specify the AdventureWorks database. Click Connect.
8. In the Subscribers page, ensure that the MELBOURNE check box is selected. Select the AdventureWorks database from the Subscription Database drop-down list. Click Next.
9. On the Distribution Agent Security page, click the button marked with four dots (...).

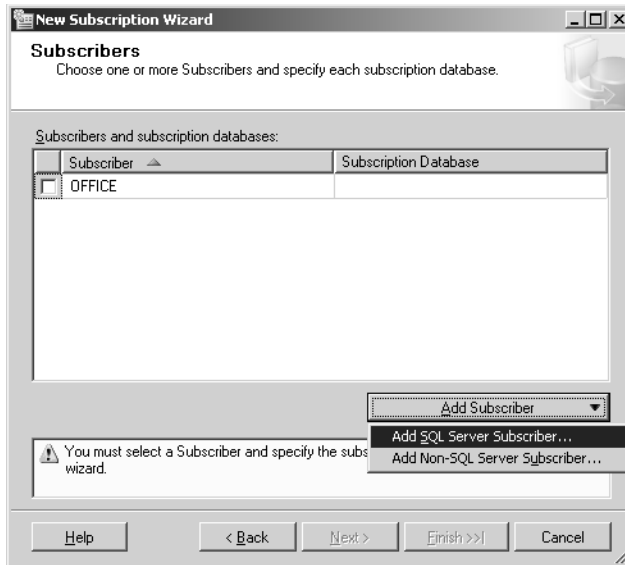


Figure 10-14 Adding a SQL Server Subscriber.

10. Configure Distribution Agent Security as shown in Figure 10-15. Click OK.

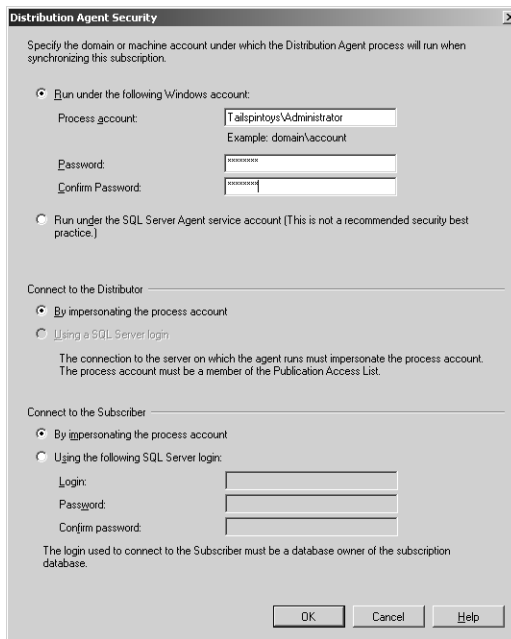


Figure 10-15 Configuring Distribution Agent Security.

11. On the Distribution Area Security page, click Next.

- On the Synchronization Schedule page, set the Agent Schedule to Run Continuously as shown in Figure 10-16. (If you have set up your test network as recommended, MELBOURNE appears in the Subscriber list rather than LONDON.) Click Next.

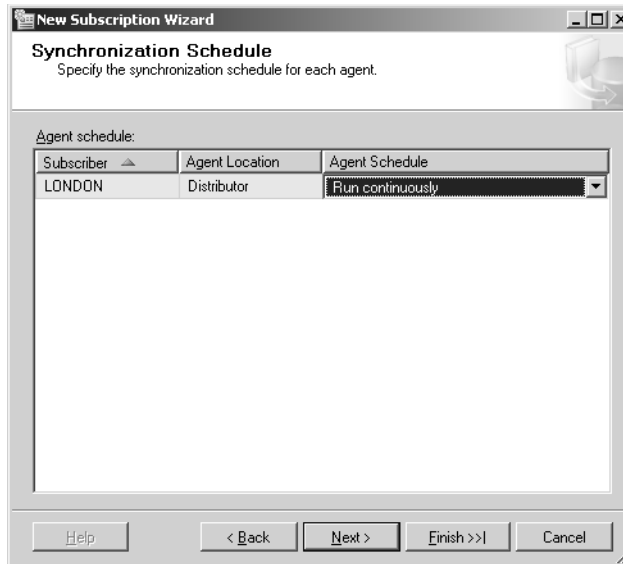


Figure 10-16 Configuring the agent to run continuously.

- On the Initialize Subscriptions page, configure the subscription to initialize immediately as shown in Figure 10-17. Click Next.

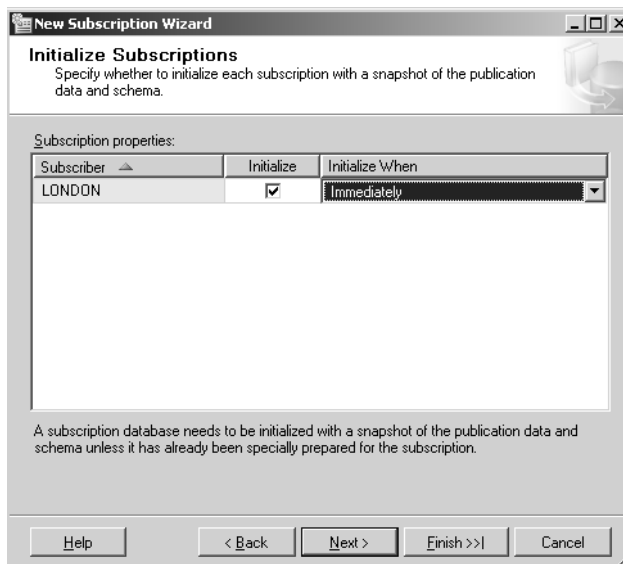


Figure 10-17 Configuring the subscription to initialize immediately.

14. On the Wizard Actions page, ensure that the Create The Subscription(s) check box is selected. Click Next.
15. Click Finish to create the subscription. When the subscription has been created, click Close.

► **Practice 3: Validating Connections and Measuring Latency**

In this practice, you use a tracer token to validate that the Publisher replicates to the Subscriber and to measure replication latency. You need to complete Practices 1 and 2 before you attempt Practice 3. If you are starting Practice 3 directly after Practice 2, SSMS is still open and connected to your member server. In this case, proceed directly to the first step in this practice. If not, log in, open SSMS, and connect to your member server as described in Practice 1.

1. If necessary, expand the server node and expand the Replication folder.
2. Expand Local Publications, right-click [AdventureWorks]: MyPub, and choose Launch Replication Monitor.
3. In Replication Monitor, select [AdventureWorks]: MyPub and then click the Tracer Tokens tab as shown in Figure 10-18.

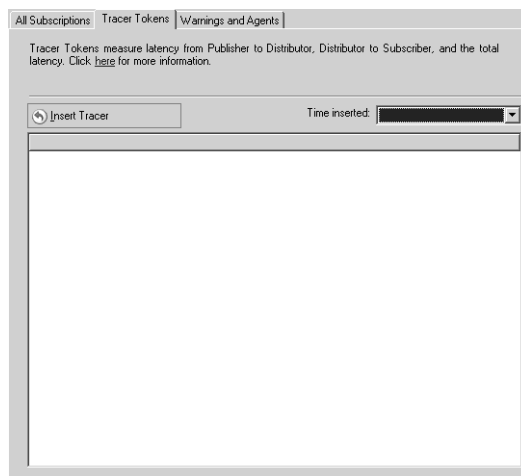


Figure 10-18 The Tracer Tokens tab.

4. Click Insert Tracer.
5. View the elapsed time for the tracer token in the following columns: Publisher to Distributor, Distributor to Subscriber, and Total Latency.

Lesson Summary

- You can implement transactional, merge, or snapshot replication in SQL Server 2005. You can set up machines as Distributors, Publishers, or Subscribers and choose the Peer-to-Peer, Central Publisher, Central Publisher with Remote Distributor, Central Subscriber, or Publishing Subscriber model.
- You can configure predefined alerts and define new alerts and warnings by using Replication Monitor. You can use Replication Monitor, SSMS, or Transact-SQL scripts to monitor replication health, latency, and failures.
- You can use tracer tokens and checksum to verify replication. The Merge Agent detects, logs, and automatically resolves conflicts, but you can also resolve conflicts manually.
- You can tune replication by optimizing server and database performance, and by tuning publication design, snapshots, subscriptions, and agent parameters.
- Replication can work with database mirroring and log shipping to provide high database availability and failover support.

Lesson Review

You can use the following questions to test your knowledge of the information in Lesson 1, “Designing a Strategy to Manage Replication.” The questions are also available on the companion CD if you prefer to review them in electronic form.

NOTE Answers

Answers to these questions and explanations of why each answer choice is correct or incorrect are located in the “Answers” section at the end of the book.

1. You are the senior DBA at a company whose organization consists of a network of retail stores distributed throughout the United States. Each retail store maintains POS transactions on a local SQL Server 2005 computer in a database table named Sales.Transactions. Customers can return goods to any of the company’s stores. As a result, each retail store requires a record of the transactions at the other stores, and this information is held in the Sales.Transactions table. On an hourly basis, the retail outlets need to replicate fresh transactions to the SQL Server 2005 server at the central office, and receive updated information on the transactions made in other stores from the central office server. Real-time replication is not a requirement. What replication configuration should you recommend?

- A. Snapshot replication between the central office and each retail store.
 - B. Transactional replication using the Central Publisher model, with the SQL Server 2005 server at the central office configured as both a Distributor and Publisher.
 - C. Merge replication between the central office and each of the retail stores.
 - D. Transactional replication using the Remote Distributor model, with the server at each retail outlet configured as a Distributor.
2. The Manufacturing.Products table on a SQL Server 2005 server contains a CHECK constraint that is used to ensure that users enter product codes in a valid format. The server is configured to publish the table. A second server is configured as a Subscriber for replication. You want to ensure that the CHECK constraint exists only at the Publisher and is not replicated to the Subscriber. What should you do?
 - A. Configure the CHECK constraint as NOT FOR REPLICATION on the table in the Subscriber database.
 - B. Configure the CHECK constraint as NOT FOR REPLICATION on the table in the Publisher database.
 - C. Reconfigure the constraint as a UNIQUE constraint on the table in the Subscriber database.
 - D. Reconfigure the constraint as a UNIQUE constraint on the table in the Publisher database.
3. Your organization's database infrastructure consists of one SQL Server 2005 SP1 server that contains a single database. All company employees regularly connect to that database. You are tasked with making the system fault-tolerant. Failover needs to occur instantly and without administrator intervention. Clients should be able to connect to the database even if the SQL Server 2005 computer experiences complete hardware failure, including a failure of all connected disks. You configure a second computer as a SQL Server 2005 server. What should you do next?
 - A. Implement a Microsoft Cluster Server (MSCS) cluster that contains the database.
 - B. Implement transactional replication.
 - C. Implement log shipping.
 - D. Implement database mirroring.

4. You have implemented merge replication between several SQL Server 2005 servers. Sometimes information can be out of date, and you suspect that merge replication is timing out before the merge can complete. How do you ensure that you receive a notification if a merge replication is about to time out?
 - A. Enable the Warn If A Merge Length For LAN Connections Exceeds The Threshold warning.
 - B. Enable the Warn If A Subscription Will Expire Within The Threshold warning.
 - C. Enable the Warn If Rows Merged Per Second For LAN Connections Is Less Than The Threshold warning.
 - D. Enable the Warn If Latency Exceeds The Threshold warning.
5. You have configured merge replication to enable databases on SQL Server 2005 servers at each of your company's branch offices to replicate to a SQL Server 2005 server at a central office, which collates the information and replicates it to all branches. A trigger at the central office generates summary reports for senior management. You do not want the trigger to replicate to the branch offices. What clause should you include in the trigger?
 - A. NOT FOR REPLICATION
 - B. ALL SERVER
 - C. INSTEAD OF
 - D. WITH ENCRYPTION
6. Your company has three automated testing areas, each of which uses a SQL Server 2005 server to store test results. You need to ensure that test results flow continuously from the testing areas to the central SQL Server 2005 server with minimum latency. What should you do?
 - A. Create a transactional publication at each of the testing areas for the test results. Create push subscriptions at each of the testing areas to the central server.
 - B. Create a snapshot publication at each of the testing areas for the test results. Create pull subscriptions on the central server for each testing area.
 - C. Create a merge publication at each of the testing areas for the test results. Create pull subscriptions on the central server for each testing area publication.

- D. Create a transactional publication at each of the testing areas for the test results. Create a pull subscription on the central server to each of the testing areas.
7. Your company's infrastructure comprises 20 SQL Server 2005 servers at remote locations, each of which subscribe to a transactional replication publication on a SQL Server at a central office. You need to check replication latency for all Subscribers. What should you do?
 - A. Configure the Warn If A Subscription Will Expire Within The Threshold warning.
 - B. Configure the Warn If Rows Merged Per Second For LAN Connections Is Less Than The Threshold warning.
 - C. Use tracer tokens to measure and record latency for each Subscriber.
 - D. Enable the Warn If Latency Exceeds The Threshold warning.

Chapter Review

To further practice and reinforce the skills you learned in this chapter, you can perform the following tasks:

- Review the chapter summary.
- Review the list of key terms introduced in this chapter.
- Complete the case scenarios. These scenarios set up real-world situations involving the topics of this chapter and ask you to create a solution.
- Complete the suggested practices.
- Take a practice test.

Chapter Summary

- You have a choice of configurations when configuring replication. You can specify a replication strategy, replication type, and replication model.
- You need to monitor replication health, latency, and failures. You can configure alerts and warnings for this purpose.
- You need to verify that replication has occurred and that data has replicated without error. You can specify whether you resolve conflicts manually or use automatic resolution.
- You can tune a wide variety of components to ensure efficient replication.
- Database mirroring and log shipping combine with replication to provide failover protection and database availability.

Key Terms

Do you know what these key terms mean? You can check your answers by looking up the terms in the glossary at the end of the book.

- article
- checksum
- database mirroring
- Distributor
- latency
- merge replication

- publication
- Publisher
- quiescing
- replication agent
- snapshot replication
- Subscriber
- subscription
- tracer token
- transactional replication

Case Scenarios

In the following case scenarios, you will apply what you've learned about replication. You can find answers to these questions in the "Answers" section at the end of this book.

Case Scenario 1: Selecting Replication Type and Model

You are a database consultant. Organizations give you their replication requirements and you advise them about what replication type and model best meets those requirements. You then guide the organizations' staff through the implementation of your recommended replication strategies. Answer the following questions:

1. Northwind Traders is a financial trading company with a central office and six large branch offices located in the United States, Brazil, France, China, Japan, and Australia. Each of these locations contains three to six SQL Server 2003 servers. A database table at the central office holds financial information, and central office employees update the table continuously. Northwind wants to replicate this table to all SQL Server 2005 servers at all its branch offices. The company operates in an exceptionally dynamic environment and requires that its branches receive the latest information in as near real time as possible. Management accepts that WAN links could limit the speed of replication but requires latencies of three minutes at the most. What replication type and model do you recommend?
2. Tailspin Toys is a franchise operation with a large number of toy and novelty stores located throughout the United States. The franchises operate independently,

but they replicate financial information from a SQL Server 2005 server in each store to a subscription database in the central office every night. This replication is already configured. Every six months the central office updates the sales catalog. Changes to the catalog are usually major, with a lot of new products added and old products removed. How should Tailspin Toys replicate this catalog?

3. Humongous Insurance has a main office and 10 branch offices. All the branch offices enter customer information and policy details into a database on a SQL Server 2005 server. A trigger on a table in the database starts an application that prints a reminder notice when a customer's policy is due for renewal. Humongous Insurance wants each branch office to be able to access information held at all branch offices. The company requires that, on an hourly basis, all database updates at branch offices replicate to the main office, and that the collated information replicates to the branch offices. Management, however, is concerned that adding information about policies in other branches will trigger reminder notices for customers other than local branch customers. What type and model of replication would you recommend, and how should Humongous Insurance reconfigure the trigger?

Case Scenario 2: Tuning Replication

You have been tasked with improving replication efficiency at Litware Inc. Litware owns a single manufacturing facility. A SQL Server 2005 server replicates production, financial, and human resource information to other SQL Server 2005 servers within the facility. Litware uses transactional replication and the Central Publisher model. Updates are not permitted at Subscribers. Answer the following questions:

1. Litware currently uses a single hardware RAID10 disk array to store databases, transaction logs, SQL Server software, and the Windows Server 2003 operating system. Currently over 75 percent of disk capacity is used and you have a budget for more hard disks. How would you partition your information?
2. You want to write a script that measures latency, and you want to configure a warning in Replication Monitor that triggers if latency exceeds a defined threshold. What stored procedures would you use, and what warning would you configure?
3. You suspect that ad hoc queries and applications that users run against the database are causing contention with replication agent activity, resulting in waits and deadlocks. What option should you configure on the publication and subscription databases to reduce this contention?

Suggested Practices

To help you successfully master the exam objectives presented in this chapter, complete the following tasks.

Design a Strategy to Manage Replication

You should carry out all the following practices:

- **Practice 1: Configure other replication types.** You followed a step-by-step procedure to implement transactional replication using the Central Publisher model. Configure other replication types (for example, merge and snapshot replication) and other models (for example, Peer-to-Peer).
- **Practice 2: Use Replication Monitor.** Use Replication Monitor until you are familiar with all the tabs and detail pages that the tool provides. Familiarity with the SQL Server 2005 tools comes only with hands-on practice.
- **Practice 3: Use a Transact-SQL query to determine latency.** Write and debug a Transact-SQL query that returns latency for transactional replication with a Subscriber. Books Online provides sample routines that can help you complete this practice.
- **Practice 4: Set the READ_COMMITTED_SNAPSHOT option to ON.** Use Transact-SQL statements to set the READ_COMMITTED_SNAPSHOT database option to ON for a publication and a subscription database.
- **Practice 5: Work with agent profiles.** Use SSMS, Replication Monitor, and Transact-SQL stored procedures to create, modify, and remove agent profiles and specify that an agent should use a profile during synchronization.

Take a Practice Test

The practice tests on this book's companion CD offer many options. For example, you can test yourself on just one exam objective, or you can test yourself on all the 70-444 certification exam content. You can set up the test so that it closely simulates the experience of taking a certification exam, or you can set it up in study mode so that you can look at the correct answers and explanations after you answer each question.

MORE INFO Practice tests

For details about all the practice test options available, see the "How to Use the Practice Tests" section in this book's Introduction.

Chapter 11

Security Strategies

As an experienced, professional database administrator (DBA), you should view database security as an ongoing activity and ensure that database systems continually meet security requirements. You need to understand the enterprise security model and how database security fits into it. You need to document and enforce procedures and requirements that are specific to the DBA's security perspective and outside the individual and database-centric viewpoint of the database developer (DBD). This chapter discusses the requirements of server-level and user-level security from a DBA's point of view.

Exam objectives in this chapter:

- Maintain a server-level security strategy.
 - Design a strategy to audit Windows account permissions.
 - Design a strategy to audit SQL Server service access.
 - Maintain a strategy to assign the appropriate minimum level of privileges.
 - Maintain an encryption strategy that meets business requirements.
 - Design a strategy to apply service packs and security updates.
 - Configure the surface area.
- Maintain a user-level security strategy.
 - Verify the existence and enforcement of account policies.
 - Verify SQL Server login authentication.
 - Verify permissions on SQL Server roles and accounts.

Lessons in this chapter:

- Lesson 1: Maintaining a Server-Level Security Strategy 585
- Lesson 2: Maintaining a User-Level Security Strategy 612

Before You Begin

To complete the lessons in this chapter, you must have completed the following tasks:

- Configured a Microsoft Windows Server 2003 R2 computer with Microsoft SQL Server 2005 Enterprise Edition SP1 as detailed in the Appendix.
- Installed an updated copy of the AdventureWorks sample database as detailed in the Appendix.

No additional configuration is required for this chapter.

Real World

Ian McLean

I've experienced network security issues from two viewpoints. As a consultant, I have advised organizations about how to set up security, harden their servers and networks, and reduce the attack surface of their systems. I've also administered security on the database servers of a large organization. I'm hardly about to decry the work of a security consultant—it's a difficult job and deservedly well paid. However, I reckon that the everyday task of a DBA concerned with security issues is even more difficult. Configuring a system securely is easier than maintaining that security. Managers and users will pay sincere lip service to security when it is being implemented, but users will want to do their jobs (quite rightly) as quickly and easily as possible, and security indisputably gets in the way. Managers will not be pleased when they realize that the humble DBA has more power on the network than they have. Most of the work of a security administrator is invisible—except when something goes wrong. So why would anyone do it? Well, actually, it's lots of fun!

Lesson 1: Maintaining a Server-Level Security Strategy

This lesson discusses the guidelines for implementing server-level security by using various authentication methods, and it explains the requirements for developing a server-level security policy. It also discusses the guidelines for creating a password policy and determining service account permissions. It explains how to select an appropriate encryption method to develop a secure communication policy, how to design a strategy to apply service packs and security updates, and how to use Surface Area Configuration Manager to minimize the exposed surface area.

After this lesson, you will be able to:

- Specify and audit Windows account permissions.
- Audit SQL Server service access.
- Ensure that SQL Server 2005 service accounts have the minimum required permissions.
- Maintain an encryption strategy that meets business needs.
- Design a strategy to evaluate and apply service packs and security updates.
- Configure the surface area of a SQL Server 2005 server.

Estimated lesson time: 75 minutes

Specifying and Auditing Windows Account Permissions

SQL Server 2005 supports both Windows and Mixed authentication modes. In Windows authentication mode, access is based on a security token assigned during successful domain (or local server) login by a Windows account, which subsequently requests access to SQL Server resources. In other words, Windows authentication mode enables users to access SQL Server using their Windows user accounts. The Mixed authentication mode allows both Windows and SQL Server authentication. SQL Server authentication relies on the verification of *credentials* that are stored and maintained by the SQL Server. With SQL Server authentication, users can log in using an account that you create and manage within SQL Server.

NOTE Service packs

The service pack level at the time of writing this book is Service Pack 1 (SP1). Unless otherwise indicated, all the information in the chapter applies to both SQL Server 2005 and SQL Server 2005 SP1.

SQL Server 2005 introduces the facility to manage SQL Server account password and lockout properties with local and domain-based group policies. If you install SQL Server 2005 on a server running Windows Server 2003 (Enterprise Edition or above), SQL Server can use Windows security policies for SQL Server authentication. This enforces restrictions on the Windows password policy, which must be at least as stringent as the SQL Server 2005 password policy for user accounts (including service and application accounts) that require access to SQL Server 2005. When you configure SQL Server to enforce Windows password policies on a SQL login identity (ID), the password must meet the following criteria:

- The password length must be at least six characters long.
- The password must contain at least three out of the four character types. Character types are uppercase alphabetic, lowercase alphabetic, numeric, and nonalphanumeric characters.
- The password cannot match any of the values: “Admin”, “Administrator”, “Password”, “sa”, or “sysadmin”; or the name of the computer hosting SQL Server; or all or part of the name of the currently logged on Windows account. Part of an account name is defined as three or more consecutive alphanumeric characters delimited on both ends by white space (a space, tab, return, and so on) or any of the following characters: , - or _#”.

NOTE SQL password restrictions and strong Windows passwords

The SQL restrictions listed are not particularly restrictive if you are using strong Windows passwords that need to be at least seven characters long and conform to complexity requirements. If complexity requirements are met, you can use strings such as “sa” or “Password” in passwords. For example, Password&7 and saLT&pepper are valid passwords. However, if Windows complexity requirements are disabled or computers that run other operating systems (for example, UNIX) require access to your databases, the password restrictions for SQL Server login apply.

Quick Check

1. Why is the password letmein22 invalid for the user account Kim_Akers?
2. Why is the password BaL00 invalid for the user account Kim_Akers?

Quick Check Answers

1. The password contains only two character types: numeric and lowercase alphabetic. It should contain at least three.
2. The password is too short.

MORE INFO CREATE LOGIN

You can use the Transact-SQL CREATE LOGIN statement to create user accounts and to specify whether password restrictions should apply. For example, CREATE LOGIN kim_akers WITH PASSWORD = 'password', CHECK_POLICY = OFF creates an account kim_akers with the weak password *password*. This applies only to SQL Server 2005 logins—Kim would not be able to log in to a Windows 2003 domain by using this password unless password complexity rules were disabled. For more information, search Books Online for “CREATE LOGIN” or access [msdn2.microsoft.com/en-us/library/ms189751\(d=ide\).aspx](http://msdn2.microsoft.com/en-us/library/ms189751(d=ide).aspx).

Determining Service Account Permissions

Every process executed within a server requires a *security context*, which is a set of activities that can be performed by specifying a valid account (user name and password) for a computer running SQL Server 2005 and its associated services. Such accounts are known as *service accounts*. You can use service accounts to define the relationship between the service, the server, and the network. By defining service account permissions, you can create an environment in which you comply with both the principle of least privilege and the business requirements that the service is required to fulfill. Use the following guidelines for determining appropriate service account permissions:

- **Identify network access needs** If you need the service to access network resources, you should create a domain user account to run the SQL Server service. Otherwise, you should define the service accounts by using a local account.
- **Apply the principle of least privilege** For example, you should not run SQL Server 2005 under the context of the Local System account or the Network Service account. These accounts have more permissions than are required for SQL Server 2005 to run successfully.
- **Create separate accounts for different services** Isolating a service helps you reduce the potential attack area in the event that a service account is compromised. Also, if you use separate accounts, you can grant specific permissions to one service without extending the scope of other services that do not require the same level of permissions.
- **Create accounts with names that are not obvious** Do not, for example, create accounts with names such as SQLServiceAccount.
- **Configure service accounts like any other regular user accounts** Do not add them to high-privilege groups, such as Administrators.
- **Give service accounts long, complex passwords** Service account passwords are retrieved from encrypted storage and do not need to be typed in each time a

service runs. They can therefore be longer and more complex than ordinary user account passwords.

- **Change the service account user name and password regularly** This is a tedious procedure because it involves reconfiguring the services that run under the context of the account. However, unless you use exceptionally long (30 characters or more) and complex passwords, you should change the account credentials to frustrate password-cracking attackers.

BEST PRACTICES Run only required Windows services

The Windows operating system and SQL Server 2005 offer comprehensive toolsets, and you might not require all these tools and services for your SQL Server environment. In such cases, you must identify which Windows services are required for your SQL Server environment. You should not install services that are not required, and you should disable installed services you do not currently use. When a service is not installed or when you disable a service, a malicious user will not be able to use or exploit the service, thus reducing the attack surface of your servers.

Auditing Windows Account Permissions

Lesson 2 of this chapter, “Maintaining a User-Level Security Strategy,” discusses auditing the account policy and permissions applied to user accounts, the relationship between account permissions and server roles, and the use of the Resultant Set of Policy (RSOP) tool. However, service and application accounts are also user accounts. Their passwords are not typed in, but are instead retrieved from secure storage. You, or your domain administrator, need to create a service account for each service you enable. It is good practice to create a separate service account for each service, although a valid argument exists for running the SQL Server service and the SQL Server Agent service under the context of the same account. The SQL Server Agent service is discussed later in this lesson.

You should record details of the services enabled, the service accounts created, and when the names and passwords in these accounts were last created in a secure encrypted file. You need to audit this information to ensure that the service accounts do not have more *privileges* than are required for the services to run, and that no service remains enabled when it is no longer required—that is, disable the service account as well as its related service. You should audit the security log in each SQL Server 2005 server to ensure there is no suspicious activity—for example, failed login attempts—associated with your service accounts.

Auditing SQL Server Service Access

Every single process executed within a server requires a security context, which is defined by specifying a valid account identity (user name and password) for a service on a computer running an application such as SQL Server 2005. As explained earlier, the accounts specified are known as service accounts. By using service accounts, you define the relationship of a service with the server and the network. By defining the correct set of permissions for service accounts, you can create an environment in which you comply with both the principle of least privilege and the business requirements. Guidelines for creating appropriate service accounts were discussed earlier in this chapter. The level of access that is permitted for a service, such as the SQL Server service, is determined by the privileges of the service account that it runs under. To audit SQL Server service access, you need to determine these privileges.

Quick Check

- What is a security context?

Quick Check Answers

- A security context is a set of activities that can be performed by specifying a valid account identity (user name and password) for a service on a computer running an application such as SQL Server 2005.

A perception exists among some database administrators that the SQL Server service does not need to be audited and safeguarded because it cannot cause any harm. This is not correct. A SQL server administrator with an sa account or an account granted the sysadmin role is authorized to run any code within the SQL Server service by using the *xp_cmdshell* or *xp_regwrite* procedure. Fortunately, however, the service does not need to run under the context of an administrative account to perform ordinary tasks. A user account that is allowed to “log in as a service” is sufficient.

The exceptions are when the SQL server is part of a cluster, when the server needs to communicate with other SQL Server servers across a network (other than for replication), or when you need to make the *xp_cmdshell* procedure available to ordinary users. Even in these cases, the privileges of the administrative account are limited by the SQL Server service policies. In general, the lower the level of the privileges granted by the SQL Server service account, the less risky is its interaction with the operating system.

The SQL Server Agent

SQL Server does not incorporate just one service. In addition to providing the SQL Server service (MSSQLServer), it includes the SQL Server Agent service (SQLServerAgent), which can perform both user-level and administrative-level tasks. The SQL Server Agent service is set to start manually by default. It might need to run in the context of an account with administrative rights if you want to permit any of the following:

- Ordinary users running operating system tasks or scripts
- Automatic SQL Server restarting in case of a breakdown
- Detection of periods of low system activity in order to perform planned tasks during these periods

If you do not require any of these features, you can configure the SQL Server Agent service to run in the context of an ordinary user account that has the right to log on as a service. Running both services in the context of the same service account simplifies administration without increasing risk, although it could complicate auditing because when the service account logs on, you do not know which service it is running.

Documentation

As part of an audit policy for SQL Server 2005 services, you need to document the following:

- SQL Server logins from your SQL Server 2005 installation
- The local Windows password policy that SQL Server 2005 applies to SQL Server logins
- The SQL Server service accounts used to start SQL Server services

You also need to monitor and record events at both the enterprise and server levels. Typically, a domain administrator configures auditing settings at the enterprise level, but he or she should grant the DBA permission to view security and application logs.

Auditing Server Logins

You can audit a service by auditing the login activity of the account under which it runs. If each service runs in the context of a separate service account, and these accounts are used for no other purpose, login statistics can give a clear indication of service activity. Even when more than one service (for example, the SQL Server service and the SQL Server Agent service) run under the context of the same account, auditing service account logins can provide valuable information.

When an account connects to an instance of SQL Server 2005 using SQL Server authentication, it supplies a user name and password that SQL Server uses to authenticate that user and determine the appropriate access rights. These user credentials are stored in SQL Server. SQL Server authentication is supported in SQL Server 2005 and has been enhanced to allow SQL Server to enforce Windows Server 2003 password policies.

When you configure the server to use the Windows authentication mode (the preferred option), users are not required to provide an additional user name and password to access the server. With either SQL Server or Windows logins, the account logs in to the SQL Server 2005 instance and this login can be audited. You configure auditing by accessing SQL Server Management Studio, connecting to the appropriate instance of the SQL Server database engine and, in Object Explorer, right-clicking the server name and then choosing Properties. You then select the Security page, and below Login Auditing, select the desired option (such as Failed Logins Only).

You can also use the same procedure to access the Security folder and then the Logins folder. Right-clicking a login and choosing Properties gives you access to the following pages:

- **General** Provides an overview of the login configuration. You can reconfigure the default database and language.
- **Server Roles** Lists the server roles. You can add or remove the login from a server role.
- **User Mapping** Lists the databases that the login can access. For a specific database, you can change the login's default schema, user identity, and assigned database roles.
- **Securables** Shows the login's current object permissions. You can change the object permissions for the login.
- **Status** Shows the login's permission to connect to the database engine, whether the login is enabled, and the login's SQL Server authentication status.

Auditing Successful and Failed Logins

Windows auditing for successful domain logins, failed domain logins, or both is normally configured by the domain administrator within group policy. However, he or she would typically grant the DBA permission to view the security log. A DBA whose account is added to the sysadmin server role can access the security and application logs on the SQL Server 2005 server. Typically, you should look for failed logins.

Repeated failed logins from the same source can indicate that a server is under attack. Failed service account logins could indicate that the service or the service account is misconfigured.

Auditing successful logins is less common. In particular, a service account could access a service and a server operating system many times in a short period. If, however, you suspect that an attacker has hijacked a service, auditing successful logins might become necessary. Take care, however. Your log files could grow very quickly and take up a lot of disk space.

Verifying Account Lockout Thresholds

After a predefined number of failed login attempts, account lockout should occur. You can audit account lockout attempts in the Windows security log. A service account does not require that a password be entered, but instead retrieves the password from secure, encrypted storage. In normal operation, therefore, service accounts are not locked out. A locked-out service account could indicate a hijack attempt and is, in any case, a serious incident. It's serious because when the service account is locked out, the service cannot run. An alert should always be configured for such an event.

Typically, an ordinary user account is created for a service account and is then given a login as a service permission and a complex password to prevent an actual user from logging in as the service. In such a case, the account lockout threshold should be the same as for any user account. If you suspect that a service account is being locked out when it should not be, or is not being locked out when it should be, you should monitor the Account Lockout event. To diagnose account lockout problems, you should also monitor the following events within the security log:

- Account Logon—Failure
- Account Management—Success
- Logon Events—Failure

In an Active Directory domain, the Audit Policy settings are located in the Default Domain policy settings. To view the Auditing policy settings (assuming you have permission to do so), open the Group Policy MMC, double-click Computer Configuration, double-click Windows Settings, double-click Security Settings, double-click Local Policies, and then double-click Audit Policy. Enable auditing for the event types listed in the previous section.

On a stand-alone server, you can view the Account Lockout setting by opening the Group Policy Object Editor MMC, double-clicking Computer Configuration, double-clicking Windows Settings, double-clicking Security Settings, double-clicking Account Policies, and then double-clicking Account Lockout Policy.

Assigning the Appropriate Minimum Level of Privileges

You can assign permissions one of three states: Grant, Deny, or Revoke (which removes existing Grant or Deny permissions). Permissions applied on higher levels imply the same states on lower levels, unless Deny (which always takes precedence) is explicitly used. Increased granularity simplifies implementing the principle of least privilege by allowing you to delegate individual tasks without granting membership of privileged server or database roles. SQL Server 2005 lets you apply some of the most commonly used permission types present in earlier versions of SQL Server—such as EXECUTE, SELECT, or TAKE OWNERSHIP—on different levels and employ inheritance capabilities. SQL Server 2005 introduces new permission types such as CONTROL, ALTER, IMPERSONATE, and VIEW DEFINITION.

- **CONTROL** This permission type is functionally equivalent to having all permissions granted to the object's owner and inherited by all sub-entities within its scope.
- **ALTER** This type lets you alter properties of an object. Depending on the scope, you can limit permission inheritance to objects of a specific type (for example, the ALTER ANY 'object_type' grants permissions to modify every instance of 'object_type' within the server or database scope). ALTER TRACE lets a user run Profiler without requiring membership in the sysadmin fixed server role.
- **IMPERSONATE** This type permits the impersonation of another user—without requiring sysadmin or database owner (DBO) privileges, as was the case in SQL Server 2000.
- **VIEW DEFINITION** This permission type gives read access to an object's *meta-data* via catalog views.

Re-Evaluating Permissions Strategy and Scope

In view of the increased permission granularity available in SQL Server 2005 and the new permissions available, you need to re-evaluate permission strategies used in earlier SQL Server releases. For new installations of SQL Server 2005, this re-evaluation should be automatic, because you are setting everything up from scratch. However, old administrator habits die hard.

If you have upgraded from SQL Server 2000 or SQL Server 7.0, you need to ensure that inherited roles and permissions, especially for service accounts, remain the optimum solutions and do not breach the principle of least privilege. Service accounts with the same privileges as the sa account—and the same blank password—are hopefully a thing of the past. However, service accounts might still have more privileges than they require. In many cases, a service might be running in the context of a user account with elevated privileges, when the only permission the account requires to run the service is “log in as a service.”

Applying the Principle of Least Privilege

The principle of least privilege has been mentioned several times already in this chapter—because it features heavily in the examination objectives. A good administrator will always strive to give users the minimum privileges they require to do their jobs, but it is harder to remember that the same principle should be applied to non-human accounts such as service and application accounts. A service account typically logs on without user intervention and works quietly in the background, causing no problems at all—until it is hijacked by a malicious attacker.

Exam Tip Always apply the principle of least privilege. This principle features heavily in the exam objectives, and you can reasonably assume that it will also feature in the exam. However, an exam question is unlikely to specifically ask you to apply the principle to a specific situation. Instead, you will probably be given a list of the tasks a user or service account needs to perform and then be asked to choose from a list of permissions and roles that could be granted to the account. Several of the answers will enable the account to perform the required tasks, but only one will also grant the account no unnecessary additional privileges. If you choose an answer that grants more than the required privileges, you will get the question wrong.

In the case of SQL Server 2005, the services for which we need to specify a service account are the SQL Server service (MSSQLServer) and the SQL Server Agent service (SQLServerAgent). Both these services can run in the context of an account with no administrative privileges, provided the following conditions apply:

- The SQL Server service is not run as a Microsoft Cluster Service (MSCS) cluster.
- Users do not have rights to run operating system commands or the `xp_cmdshell` procedure.
- The SQL Server Agent service is not required to restart the SQL Server in case of a breakdown.

- The SQL Server Agent service does not need to detect periods of low server activity.
- The SQL Server service communicates with other servers in the local area network only if it is overtly given the required rights.

Under these circumstances, the principle of least privilege is satisfied when the account under which these services run is configured as a domain user account with no special rights in the domain. Also, the account should not have any rights to log in locally as a batch job or a service on other computers. It should not have the right of network communication with other domain member servers.

To give such an account the minimum privilege it requires to run the service, your domain administrator needs to create an organizational unit (OU) and block permissions inheritance. (Policies set to No Override will still apply.) The administrator then places the server on which SQL Server 2005 runs within the OU, as well as other servers with which the SQL Server services need to communicate over the network. Next he or she creates a group policy object (GPO) specifically to apply policies to that OU. The service account should be allowed to log in as a service in the OU and can be accessed from the network.

Placing the servers in an OU enables you to delegate permissions for that OU without granting domainwide permissions. If the servers are not in an OU, the only way you can enable the database administrators to grant Windows permissions is to give them this power over the entire domain. This violates the principle of least privilege.

Maintaining an Encryption Strategy

SQL Server 2005 introduces functions that allow us to encrypt and decrypt data within Transact-SQL statements. You can use these functions to encrypt sensitive data, such as credit card details, before storing it in the database so that if the security of the database is compromised, sensitive data remains safe.

NOTE Transact-SQL or .NET assembly

Because .NET has more powerful methods for manipulating strings and byte streams, you can perform complex encryption processing more easily in a .NET assembly. However, .NET *cryptology* relies heavily on unmanaged code and, where possible, you should perform simpler tasks within Transact-SQL. This approach has the advantage of not opening up the database to any potential security vulnerabilities that could arise from misuse of the .NET methods.

SQL Server 2005 provides four pairs of encryption/decryption functions. The encryption techniques that each encryption/decryption pair uses are as follows:

- **Password encryption** This technique provides the simplest and weakest type of encryption. The data is encrypted with a user-supplied password string, with no checks on the strength of the chosen password.
- **Symmetric key encryption** This technique uses a single key for both encryption and decryption. SQL Server 2005 provides two sets of functions for symmetric key encryption: using a user-provided password as the key, and using a key stored as an object in SQL Server.
- **Asymmetric key encryption** This technique provides the most secure form of encryption because it uses different keys for encrypting and decrypting the data. However, it is slower than weaker forms of encryption, and it should not be used for large volumes of data unless security is all-important and performance is not an issue.
- **Certificate encryption** Certificates are issued by a trusted certificate authority (CA), and they are used to provide proof of the identity of the author of code or the sender of data. They are widely known from their use in Secure Sockets Layer (SSL), which is used in secure HTTPS. SQL Server 2005 can issue its own certificates.

Encryption Hierarchy

Before we look at how to create certificates, symmetric keys, and asymmetric keys, we need to discuss the encryption hierarchy of SQL Server 2005. First, the service master key is created when a SQL Server instance is installed, and it is encrypted with the Windows Data Protection Application Program Interface (DPAPI) using the password of the Windows service account under which the key was created.

This key exists at the server level, but each database can also contain a database master key used to encrypt the private keys of certificates and asymmetric keys stored in the database. The database master key is a symmetric key encrypted using the Triple DES (Data Encryption Standard) algorithm with the password supplied when the key is created.

Below the database master key in the encryption hierarchy are the certificates and asymmetric keys stored in the database, whose private keys are encrypted with the database master key. These can be used to directly encrypt data, or to encrypt symmetric keys stored in the database to provide a higher level of security than simple

symmetric key encryption. Symmetric keys form the bottom level of the encryption hierarchy and can be used to secure data or other symmetric keys.

Encryption with a User-Supplied Password

The simplest and weakest form of encryption/decryption uses a key supplied by the user. The advantage of this approach is that because the password is not kept on the system, there is no need to store it secretly. However, a user-supplied password is typically sent across the network more often than a stored password, and this requires further encryption, which could affect performance. The main disadvantages, however, are that no control exists over how securely the user stores his or her password and that SQL Server 2005 does not enforce strong user-supplied passwords. Unless the front-end application validates the password when it is chosen, the user can (and probably will) choose a noncomplex, insecure password. For added security, you should use encryption with a user-supplied password in conjunction with another encryption method.

The *EncryptByPassPhrase()* function encrypts data with a user-supplied password. This function takes as its parameters the value to be encrypted and the password, both of which can be either string literals or values of a string type (char, nchar, wchar, varchar, or nvarchar). The *DecryptByPassPhrase()* function then decrypts the data, taking as parameters the password and the cipher text. The cipher is salted with an arbitrary value so that the cipher text will be different each time you encrypt a message. This approach does not, however, affect the decryption process.

Creating Symmetric Keys

The next most simple (and least secure) method is to encrypt the data with a symmetric key stored in the database. Because the same key is used both for encrypting and decrypting data, both sides of an encrypted dialog conversation need to pass the symmetric key securely, as both sides need the same key. If, however, a SQL Server uses a symmetric key to encrypt data immediately before it is stored in a database and decrypt it immediately after it has been extracted, the key does not need to be sent across the network. The CREATE SYMMETRIC KEY statement creates a symmetric key in a database, for example:

```
CREATE SYMMETRIC KEY MyCreditCardKey  
WITH ALGORITHM = DESX  
ENCRYPTION BY PASSWORD = 'ghf@BR*hJ98R';
```

Symmetric key encryption can use the DES, TRIPLE_DES, RC2, RC4, DESX, AES128, AES192, and AES256 algorithms. In addition to (or instead of) encrypting the key with a password, you can choose to use a certificate or another symmetric key. If the key is a temporary key, you do not need to encrypt it. You can use the WITH clause to include two other options: DERIVED_FROM derives the key from a user-supplied password, and IDENTIFIED_BY lets you specify a phrase to identify data encrypted with the key.

Creating Asymmetric Keys

Asymmetric key encryption uses a private key, which is stored locally and kept secret, and a related public key. Data encrypted with the public key is decrypted with the corresponding private key, and vice versa. The private key never needs to be transmitted over the network, because data is encrypted with the intended recipient's public key and only the intended recipient can decrypt it with his or her private key.

This solution does not provide any guarantee of the sender's identity because anyone can encrypt data with a public key. Two sets of keys are therefore necessary. A sender needs to encrypt data (the signature) with his or her private key (which proves sender identity), and then encrypt the cipher text with the recipient's public key. The recipient then decrypts the cipher text with his own private key and verifies the signature using the sender's public key.

Quick Check

- A database application transmits encrypted data that only Kim Akers is able to read. Asymmetric key encryption is used. How does the application encrypt the data, and how does Kim decrypt it?

Quick Check Answers

- The application encrypts the data with Kim's public key. Kim decrypts it with her private key.

Asymmetric key encryption provides good security but poor performance, and it is seldom used for large quantities of data. In practice, you should use this method in conjunction with other encryption techniques. For example, you could create a temporary symmetric key, transmit that using asymmetric key encryption, and then encrypt the rest of the conversation with the symmetric key.

The CREATE ASYMMETRIC KEY statement has two forms. You can create a new asymmetric key pair from scratch using a specified algorithm, or you can load one

from an existing file. To create a new asymmetric key, you use the WITH ALGORITHM clause, for example:

```
CREATE ASYMMETRIC KEY AssmKey
WITH ALGORITHM = RSA_1024
ENCRYPTION BY PASSWORD = 'ghf@BR*hJ98R';
```

The possible algorithms are RSA_512, RSA_1024, and RSA_2048.

You can also create an asymmetric key pair from an existing key file, signed assembly file, or .NET assembly. For example, you can create an asymmetric key from a key file generated with the sn.exe command-line utility:

```
CREATE ASYMMETRIC KEY AssmKey
FROM FILE = 'C:\SqlServer2005\TK70-444\Encrypt\mykey.key'
ENCRYPTION BY PASSWORD = 'ghf@BR*hJ98R';
```

Creating Certificates

Certificates are used to identify a particular person or organization and to secure Web sites that use the SSL protocol. They contain a public key and additional data, such as information about the owner of the key, the start date, and expiration date. Certificates are issued by a CA such as VeriSign, Thawte, and GlobalSign. A CA performs a background check on anyone who applies for a certificate to ensure that all the information the applicant supplies is correct, and a certificate issued by one of these authorities will be trusted on most systems. However, you can also issue your own certificates using Certificate Services in Windows 2000 Server or Windows Server 2003. Such certificates are generally trusted within your own organization, but not normally by anyone outside it.

SQL Server 2005 can also issue certificates, which are used to encrypt data and to create HTTPS endpoints for secure Web services. The CREATE CERTIFICATE Transact-SQL statement enables you to create a new certificate from scratch or load one from an existing certificate file, .NET assembly, or assembly file.

If you want to create a new certificate, you need to specify its start date, expiration date, subject (the server that issued the certificate), and (optionally) the password used to encrypt the private key associated with the certificate, for example:

```
CREATE CERTIFICATE MyCertificate ENCRYPTION BY PASSWORD = 'ghf@BR*hJ98R'
WITH START_DATE = '08/12/2006',
EXPIRY_DATE = '08/12/2007',
SUBJECT = 'caserver.adventure-works.com';
```

If you omit the `ENCRYPTION BY PASSWORD` clause, SQL Server encrypts the certificate with the master key for the current database. This action is required in certain situations, such as when creating a certificate to authenticate a Service Broker endpoint.

MORE INFO Service Broker

Microsoft SQL Server 2005 Service Broker is part of the database engine. It provides a message-based communications platform that enables independent application components to perform as a functioning whole. Service Broker helps you scale your application to accommodate the amount of traffic the application receives.

SQL Server 2005 uses Service Broker *endpoints* for Service Broker communication outside of the SQL Server instance. A Service Broker endpoint configures SQL Server to send and receive Service Broker messages over the network by listening on a specific TCP port number. By default, an instance of SQL Server does not contain a Service Broker endpoint. You must create a Service Broker endpoint to send or receive messages outside the SQL Server instance. An instance can contain only one Service Broker endpoint.

For more information about Service Broker, refer to SQL Server 2005 Books Online or access [msdn2.microsoft.com/en-US/library/ms166043\(SQL.90\).aspx](http://msdn2.microsoft.com/en-US/library/ms166043(SQL.90).aspx).

You can also specify a file from which to load the private key. If you want the certificate to be made available for initiating Service Broker conversations, you need to add the following clause:

```
ACTIVE FOR BEGIN_DIALOG = ON
```

Applying Service Packs and Security Updates

Some SQL Server 2003 server maintenance activities—such as backups, database service packs, and operating system security updates—are typically regarded as routine and repetitive, and not requiring DBA interaction. This premise is false. Although you might not be the person who performs the upgrades to the production network, you must evaluate the impact of applying service packs and security updates on your environment. You need to read the documentation associated with each service pack or security update and analyze the overall impact on your systems. In all but the smallest organizations, you should have a preproduction environment similar to the production environment to test service packs and security updates. You should schedule the testing of security updates to fit with system activities.

In most cases, the most recent code is the most secure because it has been analyzed for improvements. However, an update can have a negative impact. To minimize uncertainty, you need to understand the changes implemented in the update. You need to deploy and test the update in your preproduction environment to analyze its

impact on your applications and its possible impact on the production environment. You must ensure that you include the performance and security implications in the tests that you conduct on the new update.

You need to document the corrections that should be made in your applications as a result of the updates. Documentation is important because if the server information is lost, you can replicate the exact environment of your system. In addition, this process helps you understand the behavior of interactions between the database server and the client applications.

Tracking software updates, service packs, and versions helps you analyze system malfunctions and the behavior of the environment before and after an update. You should consider the use of tools such as Windows Server Update Services (WSUS) within your preproduction environment. Where possible, use the same tools that are employed by domain and server administrators in the production environment.

Windows Server Update Services

WSUS uses several components to enable the automation of official Microsoft updates. These include the following:

- **Microsoft Update** WSUS components connect to the Microsoft Update Web site to obtain updates to Microsoft products.
- **Windows Server Update Services server** The WSUS server provides the features that administrators need for managing and distributing updates through a Web-based tool, which can be accessed from Microsoft Internet Explorer on any computer running Windows in the corporate network. In addition, a WSUS server can be the update source for other WSUS servers within the organization. In a WSUS implementation, at least one WSUS server in the network must connect to Microsoft Update to get available updates.
- **Automatic Updates** Automatic updates is the client computer component built into the Windows 2000 with SP3, Windows XP, and Windows Server 2003 operating systems. Automatic Updates enables both server and client computers to receive updates from Microsoft Update or from a server running WSUS.

Scheduling Server Reboots

When you have tested the updates in your preproduction environment, you should be involved with deployment in the production network. You need to coordinate with other IT employees who administer the domain and the network. The deployment

team needs to communicate to the users that a particular server might not be available at all times. The team should also be ready to roll back the update in case an unidentified event occurs in the production environment.

When an organization offers its users access to data 24 hours per day, 7 days per week, problems can arise because of the inability to change applications or implement changes that interrupt business processes. Identifying such issues helps you create a policy that is compliant with the business and technical requirements of the organization. In such an environment, you should group key servers—such as SQL Server 2005 servers—in failover clusters, and it is necessary for you to schedule the reboots so that an active cluster node is always available to service user requests. Chapter 4, “Disaster Recovery,” discusses clustering in more detail.

Backing Up the Database

When you make any significant changes to your SQL servers, such as security or service pack updates, you want to make sure that no data is lost if something goes wrong. Backing up your databases and transaction logs ensures that you can restore data to the point of failure. In addition to backing up your data before making any significant changes, you should also implement regular backups to ensure your data is recoverable if your database is damaged through unforeseen events. Chapter 4 discusses disaster recovery strategies in depth. However, a brief reminder of the procedures to implement a database backup strategy is appropriate here.

You can create a job that performs a database backup by using either stored procedures or SQL Server Management Studio (SSMS). For the purposes of this discussion, let’s use SMSS. Instead of manually backing up your databases within SSMS, you typically automate backups by scheduling jobs. To schedule a job, you must start the SQL Server Agent service.

By default, SQL Server Agent is not started, so typically you first need to connect to the SQL Server database engine and start it. If the service is not running, Object Explorer displays the node as “SQL Server Agent (Agent XPs disabled).” Right-clicking the SQL Server Agent node and choosing Start starts the service.

Expanding the node causes Object Explorer to display the Jobs, Job Activity Monitor, Alerts, Operators, Proxies, and Error Logs nodes. To create a job, right-click the Jobs node and choose New Job. This launches the New Job dialog box. On the General page, you can enter metadata such as the job name and the owner. On the Steps page, you can add the steps required to do the job—in this case, “Perform Backup.” This

accesses the New Job Step dialog box, where you can enter the Transact-SQL script for backing up the database, for example:

```
BACKUP DATABASE Mydatabase to DISK='C:\SQLserver2005\Backups\Mydatabase.bak'
```

You then select the Schedule tab and define the schedule. For any schedule that you specify, the SQL Server Agent service carries out the job as scheduled. You can, however, right-click any job and choose Start Job to start the job immediately.

Configuring the Surface Area

A software package has a smaller *surface area* if there are fewer ways to attack it. You reduce the surface area by, for example, closing ports that do not need to be open, closing APIs that don't need to be running, and disabling protocols and services that are not required. SQL Server 2005 enables you to explicitly manage its surface area with the SQL Server Surface Area Configuration tool.

The SQL Server Surface Area Configuration tool presents you with a brief explanation of the surface area concept and provides hyperlinks that let you start the individual surface area configuration tools: Surface Area Configuration For Services And Connections, and Surface Area Configuration For Features. Typically, you run the SQL Server Surface Area Configuration tool immediately after installing SQL Server 2005 because the default SQL Server 2005 installation has most features disabled in the interest of increasing security. You should enable only the features you need, and then use the tool at regular intervals to ensure that any features you no longer require are disabled.

Configuring Services and Connections

Typically, you access the Surface Area Configuration For Services And Connections tool first, because after the initial default installation, SQL Server 2005 cannot communicate with any other computers on the network. The Remote Connections node of this tool lets you enable TCP/IP or named pipes connections (or both). Unless you have a specific requirement for named pipes, you should enable only TCP/IP, because TCP/IP does not require you to open as many ports in your firewall.

The Service node under each component lets you selectively enable or disable the various services that collectively make up SQL Server 2005. The list of services that the tool displays varies, depending on the services you have installed. These services include the following:

- Analysis Services

- Database Engine
- Full-Text Search Service
- Integration Services Service
- MSSQLServerADHelper Service
- Notification Services Service
- Reporting Services Service
- SQL Server Agent Service
- SQL Server Browser Service
- SQL Server Writer Service

Configuring Features

After you enable the services you require, you can use the Surface Area Configuration For Features tool to enable or disable specific features. As in any security configuration, you need to strike the correct balance between functionality and safety. For example, the *xp_cmdshell* extended stored procedure offers system administrators the facility to execute operating system commands from within Transact-SQL batches, but enabling this feature allows anyone who gains access to an administrative account to attack the entire server operating system. If you do not specifically require *xp_cmdshell*, disable it. The list of features that the Surface Area Configuration for Features tool presents depends on the services you install and enable on your SQL Server 2005 server. The features you can manage with this tool include the following:

- Database Engine Features
 - **Ad Hoc Remote Queries** Lets you use OPENROWSET and OPENDATASOURCE
 - **CLR Integration** Lets you use stored procedures and other code written with the .NET common language runtime
 - **Database Mail** Lets you use the Database Mail system to send e-mail from SQL Server
 - **HTTP Access** Enables HTTP endpoints, which allow SQL Server to accept HTTP connections
 - **OLE Automation** Enables the OLE automation extended stored procedures
 - **Service Broker** Enables Service Broker endpoints
 - **SMO and DMO** Turns on Server Management Objects and Distributed Management Objects

- ❑ **SQL Mail** Lets you use the older SQL Mail syntax for sending e-mail from SQL Server
- ❑ **Web Assistant** Enables the Web Assistant for automatic output to Web pages
- ❑ **xp_cmdshell** Turns on the *xp_cmdshell* extended stored procedure
- Analysis Services Features
 - ❑ **Ad Hoc Data Mining Queries** Allows Analysis Services to use external data sources via OPENROWSET
 - ❑ **Anonymous Connections** Allows unauthenticated users to connect to Analysis Services
 - ❑ **Linked Objects** Enables dimensions and measures to be linked between instances of Analysis Services
 - ❑ **User-Defined Functions** Allows user-defined functions to be loaded from COM objects
- Reporting Services Features
 - ❑ **HTTP and Web Service Requests** Allows Reporting Services to deliver reports via HTTP
 - ❑ **Scheduled Events and Report Delivery** Enables the “push” delivery of reports

In new SQL Server 2005 installations, most of these features are disabled by default. In SQL Server 2000, by contrast, most features were enabled. A DBA accustomed to administering earlier versions of SQL Server might be tempted to turn everything back on. One of the main advantages that SQL Server has traditionally offered is its large number of powerful features. There is nothing wrong with taking advantage of powerful features—but only if you need them. If you do not need a feature, it is safer to disable it. The SQL Server Surface Area Configuration tool lets you enable any feature quickly and easily if you need it, and disable it when you have finished the task for which it was required.

PRACTICE Using the SQL Server Surface Area Configuration Tool

In the following practice session, you use the SQL Server Surface Area Configuration tool to enable the protocols, services, and features you require and to minimize the possibility of attack by disabling anything that is not required.

► **Practice: Configuring Protocols, Services, and Features**

In this practice, you check that the database service is running and enable TCP/IP for local and remote database engine connections. You then start the analysis service (unless it is already running) and specify local and remote connections. You check whether the SQL Server Agent service is running and start it if it is not. You leave the full text search, integration services, and SQL Browser services at their current settings. You then configure the features, disabling all features (or checking whether they are disabled), except the linked objects feature, for which you enable links to other instances. If you have previously configured protocols, services, or features, some of the default settings described in this practice might be altered.

1. Log in to your domain at your member server by using an account that has been added to the sysadmin server role. If you are using virtual machine software, log in to your domain and connect to your member server.
2. From the Programs menu, select Microsoft SQL Server 2005, Configuration Tools, and then SQL Server Surface Area Configuration. The tool starts, as shown in Figure 11-1.

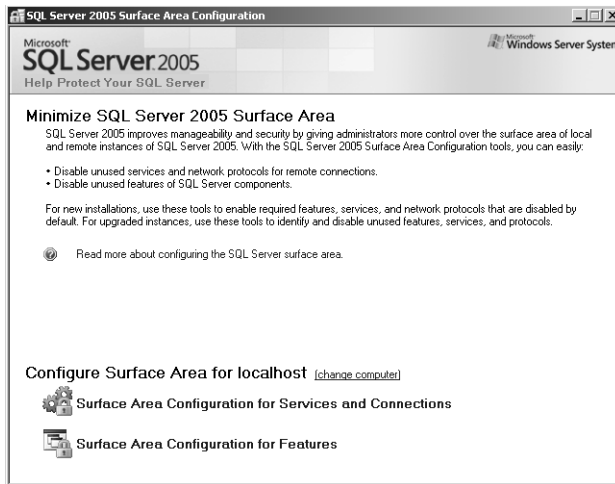


Figure 11-1 The Surface Area Configuration tool.

3. Click the Surface Area Configuration For Services And Connections link.
4. Fully expand the left-hand pane, and select Service under Database Engine. Check whether the service is running, and start it if it is not.
5. Under Database Engine, select Remote Connections, and then select Local And Remote Connections. Select the Using TCP/IP Only option, as shown in Figure 11-2.

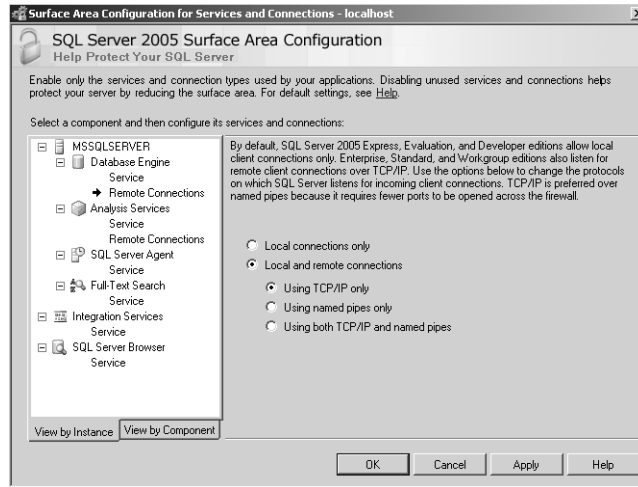


Figure 11-2 Selecting TCP/IP for local and remote connections.

6. If a Connection Settings Change Alert box appears, warning you that you need to restart the database engine service before any connection changes take place, click OK and then continue the configuration of protocols and services.
7. Under SQL Server Agent, select Service. Check whether the service is running, and start it if it is not.
8. Click OK. This returns you to the initial page.
9. Click the Surface Area Configuration For Features link.
10. In the left-hand pane, Database Engine should be expanded by default. If not, expand it. Expand Analysis Services.
11. Select Ad Hoc Remote Queries, and ensure OPENROWSET and OPENDATASOURCE support are not enabled, as shown in Figure 11-3.
12. Select CLR Integration. Check to ensure it is not enabled.
13. Select DAC. Check to ensure that remote DAC is not enabled.
14. Select Database Mail. Check to ensure that Database Mail stored procedures are not enabled.
15. Select Native XML Web Service. By default, the instance should have no endpoints.
16. Select OLE Automation. Check to ensure it is not enabled.
17. Select Service Broker. By default, it should not have an endpoint.
18. Select SQL Mail. Check to ensure that SQL Mail stored procedures are not enabled.

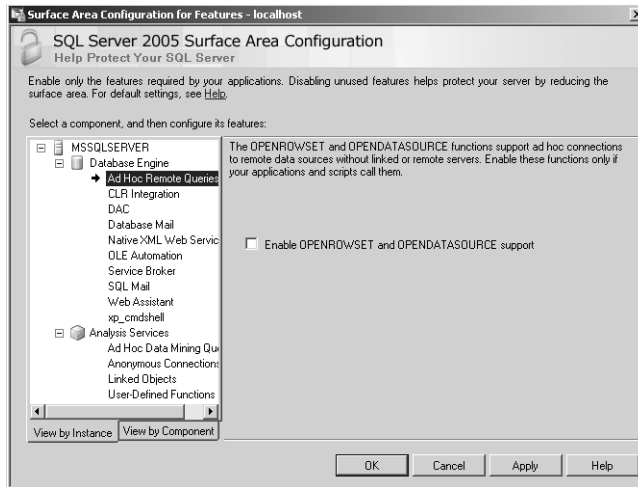


Figure 11-3 Configuring the Ad Hoc Remote Queries feature.

19. Select Web Assistant. Check to ensure it is not enabled.
20. Select xp_cmdshell. Check to ensure it is not enabled.
21. Select Ad Hoc Data Mining Queries. Check to ensure they are not enabled.
22. Select Anonymous Connections. Check to ensure they are not enabled.
23. Select Linked Objects. Select the Enable Links To Other Instances option as shown in Figure 11-4.

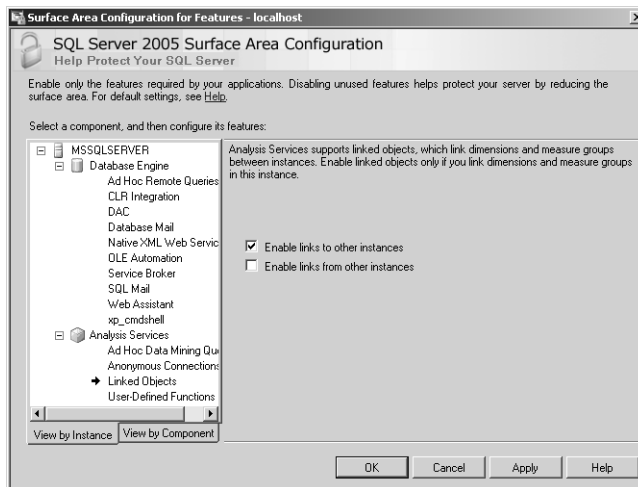


Figure 11-4 Enabling links to other instances.

24. Select User-Defined Functions. Check to ensure that the loading of user-defined COM functions is not enabled.

25. Click OK. This returns you to the initial page.
26. Click the Surface Area Configuration For Services And Connections link.
27. Stop and restart the database engine service.
28. Click OK. This returns you to the initial page.
29. Close the tool.

Lesson Summary

- Service accounts should meet the same security criteria as ordinary user accounts, although typically they have longer and more complex passwords. A system account should not be granted elevated privileges or placed in a privileged group unless there is a specific requirement to do so.
- You can audit SQL Server access by auditing service account logins.
- You can use SQL Server Management Studio to configure SQL Server auditing.
- You should apply the principle of least privilege to both service accounts and ordinary user accounts.
- The four encryption/decryption methods used by SQL Server are: password, symmetric key, asymmetric key, and certificate.
- You should test service packs and security updates on a preproduction network before they are installed on a production network. Where installation requires a reboot, you should arrange for this to be scheduled so that it causes the minimum disruption to users.
- An SQL Server 2005 server should present the smallest possible surface area for attack. You can configure surface area by using the Surface Area Configuration tool.

Lesson Review

You can use the following questions to test your knowledge of the information in Lesson 1, “Maintaining a Server-Level Security Strategy.” The questions are also available on the companion CD if you prefer to review them in electronic form.

NOTE Answers

Answers to these questions and explanations of why each answer choice is correct or incorrect are located in the “Answers” section at the end of the book.

1. The SQL Server service runs under the context of a service account with the user name PaschkeDorena. The server name is DataServer01. Which of the following is a valid password for this account?
 - A. P@\$s4
 - B. PaschkeDorena
 - C. breakfast14
 - D. U&&peK@2378
2. What SQL Server 2005 tool do you open to access the Logins folder by using Object Explorer?
 - A. SQL Server Management Studio
 - B. SQL Server Surface Area Configuration
 - C. SQL Server Configuration Manager
 - D. SQL Server Profiler
3. You are a domain administrator, and you also carry out database administration tasks. You want to delegate control of the SQL Server member servers to other database administrators. What is your initial step?
 - A. Put all the SQL Server member servers in a Windows security group.
 - B. Put all the SQL Server member servers in an OU.
 - C. Grant all the database administrators' accounts membership of the sysadmin role on the SQL Server member servers.
 - D. Put the SQL Server member servers into an MSCS cluster.
4. Which of the following encryption algorithms can you specify when generating an asymmetric key pair? (Choose all that apply.)
 - A. RSA_512
 - B. DES
 - C. RSA_1024
 - D. RSA_2048
 - E. TRIPLE_DES
5. Your SQL Server 2005 member servers obtain service packs and security updates from a WSUS server. What could the WSUS server access to obtain these service packs and security updates? (Choose all that apply.)

- A. A CA server
 - B. A WSUS server
 - C. VeriSign
 - D. Windows Update
 - E. Microsoft Update
6. You are using the Surface Area Configuration tool to configure local and remote connections for the database engine on a SQL Server 2005 member server. Which protocol or protocols should you enable and why?
- A. Named pipes because it is the standard SQL Server protocol for remote connections.
 - B. Both TCP/IP and named pipes because having both protocols enabled will provide faster network communication.
 - C. TCP/IP because it requires fewer ports to be opened on the firewall than does named pipes.
 - D. Named pipes because it is faster and more efficient than TCP/IP.

Lesson 2: Maintaining a User-Level Security Strategy

This lesson discusses the guidelines for implementing user-level security by developing and maintaining a user-level security strategy. It discusses the guidelines for creating a password policy and ensuring that it is properly applied. It explains how to ensure that strong passwords are enforced for SQL Server logins. It covers the use of `sys.sql_logins` and compares SQL Server and Windows authentication modes. The lesson discusses how to verify logins associated with server and database roles and logins assigned to databases. It also describes how you verify logins when multiple users are assigned to the same credential. Finally, it discusses permissions associated with roles and accounts, and when you would delete or disable user accounts.

After this lesson, you will be able to:

- Specify a password policy and ensure it is properly applied.
- Ensure that strong passwords are enforced for SQL server logins.
- Use `sys.sql_logins`.
- Compare SQL Server and Windows authentication modes.
- Verify permissions on SQL Server roles and accounts.
- Delete or disable unused accounts.

Estimated lesson time: 60 minutes

Verifying the Existence and Enforcement of Account Policies

Users typically log in to an Active Directory domain and, if you have configured SQL Server to use Windows integrated authentication, the same security token allows them to log in to SQL Server. If the server is not in a domain, the credentials the user supplies to log in to the Windows operating system are also used for SQL Server login. If SQL Server authentication is used, the user must provide separate credentials for SQL Server login.

When a user account has successfully authenticated with SQL Server, the user credentials are then used to access the required SQL Server instance, schema, and database. A user cannot access a database unless granted permission to do so. Strong password and account lockout policies protect SQL Server and the databases it manages from unauthorized access.

Specifying a Strong Password Policy

Before considering the permissions that should be granted to domain user and service accounts, you first need to specify a password policy that protects your valuable and sensitive data from unauthorized access. In general, the domain administrator rather than the DBA configures such a policy, but the DBA has influence in formulating it and determining that it is correctly configured. Windows security policies for domain user accounts (including service and application accounts) specify the following:

- **Password Complexity** This setting ensures that passwords meet a set of complexity requirements—for example, they should include both alphanumeric and nonalphanumeric characters.
- **Password Length** This setting specifies the minimum number of characters that a valid password can contain. Passwords with fewer characters are not accepted.
- **Password History** This setting ensures that users cannot re-use recent passwords when their passwords expire and they need to change them.
- **Password Age** This setting specifies the maximum and minimum allowed age for a password—that is, how long a password will be valid for before the user is forced to change it, and the shortest time that must elapse after a user has specified a password before he or she can change it.
- **Account Lockout Threshold** This setting specifies how many chances a user gets to enter a password correctly before his or her account is locked.
- **Account Lockout Duration** This setting specifies how long an account will be locked after a user has exceeded the account lockout threshold.

NOTE Minimum allowed password age

Although the purpose of the maximum allowed password age is self-evident, many people find the minimum allowed password age puzzling. Suppose my password is MyP@ssw0rd1, and I want to keep it. Suppose also that the password history setting is six. When required to change my password, I would change it to MyP@ssw0rd2. If no minimum allowed password age was specified, I could immediately change it to MyP@ssw0rd3, and then to MyP@ssw0rd4, all the way through to MyP@ssw0rd7. I am now able to change my password to MyP@ssw0rd1 without violating any password rule. If, however, I need to wait for (say) 24 hours between each change, I would probably decide not to carry out this procedure.

When specifying password policy, you need to consider password length, strength, changes, and storage. The following Microsoft recommendations are generally

accepted as industry standards:

- A user account password should ideally be more than nine characters long. The minimum recommended length for a SQL Server login's password is six characters. The minimum recommended length for a secure Windows login's password is seven characters.

NOTE Password length

More than nine characters is a recommendation, and you should treat it as such. Some documents recommend seven or more characters. Some high-security organizations insist on a minimum of 13 characters. Six characters is a minimum recommendation, and again you should treat it as such.

- A user account password should include both uppercase and lowercase letters, numbers, and permitted nonalphanumeric characters. For a SQL Server login, the password must contain at least three out of the four character types.
- Application account passwords should be longer than user account passwords. Typically, application accounts are not updated as frequently as user accounts.
- Periodic password changes help to keep the system secure. If an attacker knows that the password is changed frequently, he or she might be deterred from trying to break it; if the attacker is trying to break a password and the password is changed, the attacker must restart the process; if an attacker has already broken a password and the password is changed, the attacker is forced to attempt to break the password again.
- Always change the passwords when moving an application from a SQL Server 2005 development environment to a test environment and from a test environment to a production environment. Otherwise, developers and testers have access to the passwords, and they can access confidential production information.
- Whenever possible, use Windows authentication to access SQL Server. This prevents transfer of credentials over a network.

Real World

Ian McLean

You need to temper the need for password security with common sense. In a high-security environment, such as a clearing bank or a military research establishment, staff accept the need for complex, 13-character passwords changed every three months, and memorizing such passwords is part of the job description.

Where security is not seen to be quite as crucial, a similar policy results in users writing their passwords down—usually on sticky notepads attached to their monitors.

Ensuring that Password Policies Are Properly Applied

No network user account—whether an ordinary user account, an account with administrative privileges (for example, a DBA account), or a service account—should be given more privileges than it requires. You need to ensure that user accounts are not placed in privileged groups such as Administrators or Domain Admins. You also need to ensure that password and account lockout policies are correctly configured. Checking the result of policy settings can be an extremely complex task, taking into account the power and complexity of group policy settings, inheritance blocking, and no-override settings, and whether local policy and domain policy conflict. Fortunately, Windows Server 2003 (and Windows 2000 Server) provides the Resultant Set of Policy (RSoP) tool.

You might not have permission to run RSoP on the production network, but you can request a copy of its output. You can and should use the tool when you are testing settings on your pre-production network.

Resultant Set of Policy

Resultant Set of Policy (RSoP) is a Windows Server Group Policy tool that you can use to check that policies, such as the account policy, have been correctly configured. RSoP is a query engine that polls existing policies and planned policies, and then reports the results of those queries. It polls existing policies based on site, domain, and domain controller, and it provides details about all policy settings that are configured by an administrator.

The Resultant Set Of Policy Wizard enables you to create an RSoP query. You can open the wizard from the Active Directory Users And Computers or Active Directory Sites And Services MMCs. You need to run the wizard at least once to create an RSoP query. When the query is complete, the wizard displays the results in the MMC RSoP snap-in. You can create many RSoP queries by adding multiple RSoP snap-ins to the MMC, but only one RSoP snap-in per query.

Administrators can define a security policy through individual security settings in Active Directory, including password and permission policies. Security settings in a GPO can also establish a security policy on a local computer or an OU containing

users, computers, or both. When there are conflicts, security settings that are defined in Active Directory always override any security settings that are defined locally. RSoP processes and displays the resulting policy for any computer or user.

RSoP also reports the scope of a GPO according to security group membership. Where an administrator implements security settings by applying a security template, RSoP verifies those settings by polling the system and displaying the resultant policy. RSoP indicates a misapplied or overridden policy setting and the policy setting's precedence.

You can use RSoP both to verify the effect of new security settings before you implement them, and to troubleshoot when security settings do not produce the result you expected.

Quick Check

- From which MMC snap-ins can you access the RSoP tool?

Quick Check Answers

- You can access the RSoP tool from both Active Directory Users And Computers and Active Directory Sites And Services.

Verifying SQL Server Login Authentication

SQL Server runs on a Windows operating system (typically, Microsoft Windows 2000 Server or Windows Server 2003). Such security mechanisms as user and service account passwords and permissions need to be implemented correctly in Windows; otherwise, SQL Server security can be compromised. You need to be familiar with the guidelines for developing a strong password policy, for determining service account permissions, and for identifying the Windows services that need to be enabled or disabled.

Attackers typically attempt to gain a basic level of privilege by exploiting security breaches. They then try to escalate the attack by seeking administrative privileges. To prevent privilege-escalation attacks, you need to specify the principle of least privilege for all Windows accounts (including service accounts).

The Principle of Least Privilege

The principle of least privilege states that you should provide the minimum access rights that users require to accomplish a task, limiting as much as possible the security privileges granted to any account.

In earlier versions of SQL Server, users needed administrative privileges to perform such tasks as creating a server trace from SQL Profiler. SQL Server 2005 permits the administrator to assign only the permissions required to perform the specified task. Also, in earlier versions of SQL Server, a user with any type of access to a database could view the entire database structure. However, when a user queries a catalog view (for example, `sys.tables`) in SQL Server 2005, the database engine checks for user permissions and shows only those objects on which the user has permissions.

In earlier versions of SQL Server, when a user accessed a database object that was owned by a DBO, the user needed an elevated privilege that exceeded his or her actual access requirement. In SQL Server 2005, you can create schemas that group database objects, and assign schema ownership only to users who need it.

Context switching can help you to apply the principle of least privilege in stored procedures. For example, if a developer needs to perform truncate table operations, you can write a stored procedure in which you switch the security context by using `EXECUTE AS`, which executes the `TRUNCATE TABLE` command. The developer then needs only execute permissions on the stored procedure.

IMPORTANT The principle of least privilege should have universal application.

The examples given describe how this principle is applied to domain user accounts when accessing databases or stored procedures. However, the principle has universal application. A domain, server, or database administrator should not perform everyday tasks, such as writing a report, while logged on with an account that has administrative privileges. Typically, you should log in with ordinary user credentials and use the `runas` command if you want to perform an administrative task. You should not give service and application accounts more privileges than required to fulfill the functions for which they are created. A solution that gives any account more privileges than required, however convenient this might seem to be in the short term, is inevitably unsound.

Ensuring that Strong Passwords Are Enforced

On a test or preproduction network, you can verify the account policy settings on the machine that is running SQL Server 2005 by opening the Local Security Settings Microsoft Management Console (MMC) snap-in from Administrative Tools/Local Security Policy (on a member server), or the Default Domain Controller Security Settings MMC snap-in from Administrative Tools/Domain Controller Security Policy (on a domain controller). To review the settings, expand the Account Policies node, select the Password Policy node, and then double-click (for example) the Minimum Password Length policy.

However, as a DBA you might not have permission on a production network to use these tools. In this case, you can use DBCC commands to verify that the account policies have been correctly configured. For example, if the minimum password length is six, you could issue the following command:

```
CREATE LOGIN MattBerg WITH PASSWORD = 'M@tt2'
```

SQL Server will reject this command with a password violation failed message, because the password does not meet the minimum password length requirement.

Using Transact-SQL to Verify SQL Server Login Authentication

You can use Transact-SQL to verify login authentication in addition to other security settings, such as encryption and database permissions. Transact-SQL catalog views and the LOGINPROPERTY Transact-SQL function provide this functionality.

Transact-SQL Security Catalog Views Security information is exposed in Transact-SQL security catalog views. You can use the following catalog views to access catalog metadata:

- Database-Level Views
 - `sys.database_permissions`
 - `sys.database_role_members`
 - `sys.database_principals`
 - `sys.master_key_passwords`
- Server-Level Views
 - `sys.server_permissions`
 - `sys.sql_logins`
 - `sys.server_principals`
 - `sys.system_components_surface_area_configuration`
 - `sys.server_role_members`
- Encryption Views
 - `sys.asymmetric_keys`
 - `sys.crypt_properties`
 - `sys.certificates`
 - `sys.key_encryptions`
 - `sys.credentials`
 - `sys.symmetric_keys`

MORE INFO Security catalog views

For more information about security catalog views, search SQL Server 2005 Books Online for Security Catalog Views (Transact-SQL) or access [msdn2.microsoft.com/en-US/library/ms178542\(SQL.90\).aspx](http://msdn2.microsoft.com/en-US/library/ms178542(SQL.90).aspx).

The most useful of these views from the point of view of verifying SQL Server login authentication is `sys.sql_logins`, which you can use to determine whether the password policy or password expiration is enforced. This view returns one row for every SQL login. Table 11-1 lists the columns returned by the `sys.sql_logins` catalog view.

Table 11-1 Columns Returned by `sys.sql_logins`

Column Name	Data Type	Description
<inherited columns>	<inherited>	Inherits from <code>sys.server_principals</code> .
<code>is_policy_checked</code>	bit	Password policy is checked.
<code>is_expiration_checked</code>	bit	Password expiration is checked.
<code>password_hash</code>	varbinary(256)	Hash of SQL login password.

A user that does not have `CONTROL SERVER` permission will see a `NULL` value in the `password_hash` column of `sys.sql_logins`. If you use Windows authentication, no password is stored in the server.

The following example grants the SQL Server login `Smacrae` permission to select a view that lists SQL Server logins. It then grants the additional permission that is required to view metadata on SQL Server logins that are not owned by the user:

```
USE AdventureWorks;
GRANT SELECT ON sys.sql_logins TO Smacrae;
GRANT VIEW SERVER STATE to Smacrae;
GO
```

To find all SQL Server logins in a server, a user with the appropriate permissions can issue the `DBCC` command:

```
select * from sys.sql_logins
```

NOTE SQL Server 2000

In SQL Server 2000, this is equivalent to:

```
select * from master..syslogins where isntgroup=0 and isntname=0
```

Quick Check

1. What are the three categories of Transact-SQL security catalog views?
2. Which Transact-SQL security catalog view is most useful for verifying SQL Server login authentication, and to which category does it belong?

Quick Check Answers

1. The three categories are database-level views, server-level views, and encryption views.
2. The `sys.sql_logins` view is most useful, and it belongs to the server-level views category.

LOGINPROPERTY (Transact-SQL) The built-in Transact-SQL `LOGINPROPERTY` function returns information about the password policy settings of a SQL Server login. The function has the following arguments:

- **login_name** The name of a SQL Server login for which login property status will be returned
- **IsLocked** Returns information that indicates whether the login is locked
- **IsExpired** Returns information that indicates whether the login has expired
- **IsMustChange** Returns information that indicates whether the login must change its password the next time it connects
- **BadPasswordCount** Returns the number of consecutive attempts to log in with an incorrect password
- **BadPasswordTime** Returns the time of the last attempt to log in with an incorrect password
- **HistoryLength** Returns the length of time the login has been tracked using the password-policy enforcement mechanism
- **LockoutTime** Returns the date when the SQL Server login was locked out because it had exceeded the permitted number of failed login attempts
- **PasswordLastSetTime** Returns the date when the current password was set
- **PasswordHash** Returns the hash of the password

The names of the properties are not case sensitive, so property names such as *HistoryLength* and *historylength* are equivalent. The values of the *PasswordHash* and *PasswordLastSetTime* properties are available on all supported configurations of SQL Server 2005, but the other properties are available only when SQL Server 2005 is running on Windows Server 2003 and both `CHECK_POLICY` and `CHECK_EXPIRATION` are enabled.

LOGINPROPERTY requires VIEW permission on the login. When requesting the password hash, it also requires CONTROL SERVER permission. *IsLocked*, *IsExpired*, and *IsMustChange* are of type int and return 1 if the login is in the specified state and 0 if it is not. *BadPasswordCount* is of type int. *BadPasswordTime*, *HistoryLength*, *LockoutTime*, and *PasswordLastSetTime* are of type datetime. *PasswordHash* is of type varbinary. If the login is not a valid SQL Server login, the function returns NULL.

The following example checks whether SQL Server login tanjaplate is locked:

```
SELECT LOGINPROPERTY('tanjaplate', 'IsLocked');  
GO
```

The following example checks whether SQL Server login kimakers must change its password the next time it connects to an instance of SQL Server:

```
SELECT LOGINPROPERTY('kimakers', 'IsMustChange');  
GO
```

Exam Tip No indication exists that the exam will require you to write or analyze complex Transact-SQL procedures. Nevertheless, you should know about the Transact-SQL functions that relate to the specified exam objectives.

Comparing SQL Server and Windows Authentication Modes

As previously mentioned in Lesson 1 of this chapter, "Maintaining a Server-Level Security Policy," SQL Server 2005 supports both Windows integrated and Mixed authentication modes. In Windows authentication mode, access is based on a security token assigned during successful domain (or local server) login by a Windows account, which subsequently requests access to SQL Server resources. The account must belong to the same Windows environment as the computer hosting SQL Server. In the Active Directory domain environment, the Kerberos protocol provides an additional level of protection. The Mixed authentication mode allows both Windows and SQL Server authentication. SQL Server authentication relies on the verification of credentials stored and maintained by SQL Server.

MORE INFO Secure by default

SQL Server 2005 is designed to be secure by default, resulting in a system with optimal security settings. The typical setup avoids installing or activating nonessential components and features that can expose the server and its data to potential attacks. For example, SQL Server Agent, Full-Text Search, and Data Transformation Services are set to manual startup. If you want to find out more about secure-by-default configuration, download the Trustworthy Computing white paper at www.microsoft.com/mscorp/twc/twc_whitepaper.mspx.

Windows integrated authentication is the preferred option. Active Directory enables you to centrally manage server-level security options—such as password policies, service accounts, and operating system rights and permissions—by using organizational units and group policy. In medium-sized or large organizations, you can save considerable time and maintain consistency in a server-level security policy by using Windows integrated security and taking advantage of Active Directory.

Problems Related to Windows Authentication You need to be aware of the problems associated with integrating server-level security with Windows Active Directory authentication methods such as Kerberos. If, for example, you need to exchange data between devices that lack the ability to participate in Active Directory (for example, compact devices) and member servers running SQL Server 2005, you cannot use Active Directory integrated authentication, and you must instead use SQL Server authentication on the stand-alone devices.

To use Kerberos mutual authentication, all SQL Server 2005 instances must have a service principal name (SPN) configured in Active Directory. If you run the SQL Server service under the local system account, which is the normal procedure, the SPN of each instance will be registered in Active Directory. If, however, you run the SQL Server service under any other account, you must use the Windows Server 2003 Resource Kit SetSPN.exe tool to configure the SQL Server SPN.

However, the major problem for most DBAs is a loss of control over the security environment. In Active Directory integrated security, logins are validated by domain controllers rather than SQL Server servers. If a SQL Server 2005 member server cannot access a domain controller for any reason, domain users cannot connect securely to the databases they require. Password policies in Active Directory are configured at the domain level, and these policies apply to SQL Server 2005 member servers.

The DBA is unlikely to have administrative rights to the domain. DBAs need to work with domain administrators, with tasks and areas of responsibility clearly defined, because administrators responsible for domain security implement the password policies needed for SQL Server. This situation can be alleviated by placing the database servers in their own OU and using group policy to apply appropriate settings to that OU.

IMPORTANT SQL Servers should have consistent security policies.

If you want to benefit from placing all your SQL Server 2005 member servers in an OU, the security requirements for all these servers should be the same. Remember, also, that many security settings (for example, password policy) are set at the domain level and cannot be overridden.

Verifying Permissions on SQL Server Roles and Accounts

Server roles are predefined server-level *security principals* that are granted a fixed set of permissions. You can add both SQL Server and Windows logins as members of a server role, in which case they inherit the role's permissions. If a user's permissions conflict with those granted by a server role of which the user is a member, those of the server role take precedence. You can assign multiple user accounts to a single server role.

Fixed server roles have a serverwide scope. Each member of a fixed server role can add other logins to that same role. The fixed server roles are as follows:

- **bulkadmin** Members of the bulkadmin fixed server role can run the BULK INSERT statement.
- **dbcreator** Members of the dbcreator fixed server role can create, alter, drop, and restore any database.
- **diskadmin** Members of the diskadmin fixed server role can manage disk files.
- **processadmin** Members of the processadmin fixed server role can terminate processes that are running in an instance of SQL Server.
- **securityadmin** Members of the securityadmin fixed server role manage logins and their properties. They can grant, deny, and revoke server-level permissions. They can also grant, deny, and revoke database-level permissions, and reset passwords for SQL Server logins.
- **serveradmin** Members of the serveradmin fixed server role can change serverwide configuration options and shut down the server.
- **setupadmin** Members of the setupadmin fixed server role can add and remove linked servers, and execute some system stored procedures.
- **sysadmin** Members of the sysadmin fixed server role can perform any activity in the server, but they do not have any additional permissions at the Active Directory domain or OU level unless these are explicitly granted. By default, all members of the local administrator's group (BUILTIN\Administrators) are members of the sysadmin fixed server role.

When you add a new login to SQL Server, you can define the server roles for that login. You should always apply the principle of least privilege to ensure that the login is implemented securely and is compliant with the security policy. Fixed server roles can be mapped to SQL Server 2005 permissions, as detailed in Table 11-2.

Table 11-2 Mapping Fixed Server Roles to Permissions

Fixed Server Role	Server-Level Permission
bulkadmin	Granted: ADMINISTER BULK OPERATIONS
dbcreator	Granted: CREATE DATABASE
diskadmin	Granted: ALTER RESOURCES
processadmin	Granted: ALTER ANY CONNECTION, ALTER SERVER STATE
securityadmin	Granted: ALTER ANY LOGIN
serveradmin	Granted: ALTER ANY ENDPOINT, ALTER RESOURCES, ALTER SERVER STATE, ALTER SETTINGS, SHUTDOWN, VIEW SERVER STATE
setupadmin	Granted: ALTER ANY LINKED SERVER
sysadmin	Granted with GRANT option: CONTROL SERVER

Members of the securityadmin fixed server role can grant both server-level and database-level permissions. In addition, all user accounts are, by default, assigned to the public role, which is granted VIEW ANY DATABASE. This does not mean that any user can view any database. User accounts need to be added specifically to databases or database views to allow users to access them and run queries against them.

Fixed server roles are defined at the server level and have permissions to perform specific server-level administrative activities. They cannot be added, removed, or changed. You can assign user accounts to server roles when you create them. You can also use the following stored procedures to add an account to a role, remove an account from a role, and verify which accounts are members of a specific role:

- ***sp_addsrvrolemember*** Adds a login account to a fixed server role
- ***sp_helpsrvrolemember*** Displays a list of the members of a fixed server role
- ***sp_dropsrvrolemember*** Removes a member of a server role

You can use SQL Server 2005 Server Management Studio to view the fixed server roles. In SSMS, you can expand a server group, expand a server, expand Security, and then select Server Roles.

Quick Check

- Which server-level permission is granted to members of the diskadmin fixed server role?

Quick Check Answers

- Granted: ALTER RESOURCES

CAUTION `sp_srvrolepermission`

The server roles available in SQL Server 2005 are the same as those available in SQL Server 2000, but the associated permissions differ. The `sp_srvrolepermission` stored procedure that is used to discover what permissions are granted to a particular server role still exists, but unfortunately it still returns the permissions of the role in SQL Server 2000, not (in SQL Server 2005 and in SQL Server 2005 SP1) the updated permissions for SQL Server 2005.

Verifying Logins Assigned to Server Roles

You can use the `sp_helprolemember` stored procedure to list and verify the accounts that are members of a database role. This procedure returns a table that contains a row for each account. Table 11-3 lists the results returned in each column of that table.

Table 11-3 `sp_helprolemember` Result Set

Column Name	Data Type	Description
ServerRole	sysname	Name of the server role
MemberName	sysname	Name of a member of ServerRole
MemberSID	varbinary(85)	Security identifier of MemberName

The following example lists the members of the serveradmin fixed server role:

```
EXEC sp_helpsrvrolemember 'sysadmin'
```

Verifying SQL User Accounts Assigned to Database Roles

Fixed database roles are defined at the database level and exist in each database. Members of the `db_owner` and `db_securityadmin` database roles can manage fixed database role membership, but only members of the `db_owner` database role (DBOs) can add members to that role. The fixed database roles are as follows:

- **db_accessadmin** Members can add or remove access for Windows logins, Windows groups, and SQL Server logins.

- **db_backupoperator** Members can back up the database.
- **db_datareader** Members can read all data from all user tables.
- **db_datawriter** Members can add, delete, or change data in all user tables.
- **db_ddladmin** Members can run any data definition language (DDL) command in a database.
- **db_denydatareader** Members cannot read any data in the user tables within a database.
- **db_denydatawriter** Members cannot add, modify, or delete any data in the user tables within a database.
- **db_owner** Members can perform all configuration and maintenance activities on the database.
- **db_securityadmin** Members can modify role membership and manage permissions.

In addition, every database user belongs to the public database role. When a user account has not been specifically granted or denied permissions, it inherits the permissions granted to the public database role.

IMPORTANT Change of behavior

In SQL Server 2005, members of the db_owner fixed database role can drop a database. This is a change of behavior from SQL Server 2000.

When you add a user account to a fixed database role, you should (as always) apply the principle of least privilege to ensure that database access is implemented securely and is compliant with the security policy. Fixed database roles can be mapped to SQL Server 2005 permissions, as listed in Table 11-4.

Table 11-4 Mapping Fixed Database Roles to Permissions

Fixed Database Role	Database-Level Permission
db_accessadmin	Granted: ALTER ANY USER, CREATE SCHEMA Granted with GRANT option: CONNECT
db_backupoperator	Granted: BACKUP DATABASE, BACKUP LOG, CHECK-POINT
db_datareader	Granted: SELECT

Table 11-4 Mapping Fixed Database Roles to Permissions

Fixed Database Role	Database-Level Permission
db_datawriter	Granted: DELETE, INSERT, UPDATE
db_ddladmin	Granted: ALTER ANY ASSEMBLY, ALTER ANY ASYMMETRIC KEY, ALTER ANY CERTIFICATE, ALTER ANY CONTRACT, ALTER ANY DATABASE DDL TRIGGER, ALTER ANY DATABASE EVENT, NOTIFICATION, ALTER ANY DATASPACE, ALTER ANY FULLTEXT CATALOG, ALTER ANY MESSAGE TYPE, ALTER ANY REMOTE SERVICE BINDING, ALTER ANY ROUTE, ALTER ANY SCHEMA, ALTER ANY SERVICE, ALTER ANY SYMMETRIC KEY, CHECKPOINT, CREATE AGGREGATE, CREATE DEFAULT, CREATE FUNCTION, CREATE PROCEDURE, CREATE QUEUE, CREATE RULE, CREATE SYNONYM, CREATE TABLE, CREATE TYPE, CREATE VIEW, CREATE XML SCHEMA COLLECTION, REFERENCES
db_denydatareader	Denied: SELECT
db_denydatawriter	Denied: DELETE, INSERT, UPDATE
db_owner	Granted with GRANT option: CONTROL
db_securityadmin	Granted: ALTER ANY APPLICATION ROLE, ALTER ANY ROLE, CREATE SCHEMA, VIEW DEFINITION

Managing Database Role Membership The *sp_addrolemember* stored procedure adds a database user, database role, Windows login, or Windows group to a database role in the current database. A database member added to a role by using *sp_addrolemember* inherits the permissions of the role. If you add a Windows-level security principal that does not have a corresponding database user, a database user will be created.

A role cannot include itself as a member, even when membership is only indirectly implied by one or more intermediate memberships, and *sp_addrolemember* cannot add a fixed database role, fixed server role, or DBO to a role. The stored procedure cannot be executed within a user-defined transaction.

Exam Tip You use *sp_addrolemember* to add a member to a database role and *sp_addsrvrolemember* to add a member to a server role. If I were looking for a good distractor when writing an examination question, I would find this fact interesting.

The following example adds the database user Smacrae to the Production database role in the current database:

```
EXEC sp_addrolemember 'Production', 'Smacrae'
```

The *sp_droprolemember* stored procedure removes a member from a database role by deleting a row from the *sysmembers* table. When a member is removed from a role, the member's account loses any permissions it has by membership in that role. Users cannot be removed from the public role, and DBOs cannot be removed from any role. The stored procedure cannot be executed within a user-defined transaction.

The following example removes the user Smacrae from the Production role in the current database:

```
EXEC sp_droprolemember 'Production', 'Smacrae'
```

Verifying Logins Assigned to Databases

You can use the *sp_helpuser* stored procedure to list the users in the current database and to obtain information about database roles and database-level principals. Used with no argument, the procedure lists all the database users. Used with a database role as its argument, it lists all the users added to that role. If you use the name of a database user as an argument, it lists all the roles to which that user has been added.

If you execute *sp_helpuser* with no arguments, it returns a row for every user or security principal in the database. Table 11-5 lists the information returned in each row.

Table 11-5 Information Returned for All Users

Column Name	Data Type	Description
UserName	sysname	Users in the current database
GroupName	sysname	Roles to which UserName belongs
LoginName	sysname	Login of UserName
DefDBName	sysname	Default database of UserName
UserID	smallint	ID of UserName in the current database
SID	smallint	User security identification number (SID)

When you specify a database role, executing the stored procedure returns a row for each member added to that role. Table 11-6 lists the information returned in each row.

Table 11-6 Information Returned for a Database Role

Column Name	Data Type	Description
Group_name	sysname	Name of the role in the current database
Group_id	smallint	Role ID for the role in the current database
Users_in_group	sysname	Member of the role in the current database
Userid	smallint	User ID for the member of the role

The information that *sp_helpuser* returns is subject to restrictions on access to meta-data. Entities on which the principal has no permission do not appear.

The following example lists all users in the current database:

```
EXEC sp_helpuser
```

The following example lists information about the user Smacrae:

```
EXEC sp_helpuser 'Smacrae'
```

The following example lists information about the db_securityadmin fixed database role:

```
EXEC sp_helpuser 'db_securityadmin'
```

IMPORTANT Obtaining information about SQL Server 2005 securables

The *sp_helpuser* stored procedure does not return information about SQL Server 2005 *securables*. You should use *sys.database_principals* for this purpose.

Verifying Multiple Users Assigned to a Single Credential

A credential is a record containing authentication information needed to connect to a resource outside SQL Server. Most credentials consist of a Windows login name and password. Credentials allow users who connect to SQL Server using SQL Authentication to connect to Windows or other resources outside of SQL Server.

After creating a credential, you can use the Login Properties (General Page) in SQL Server Object Explorer to map it to a login. You can map a single credential to multiple SQL Server logins, but you can map a SQL Server login to only one credential. System credentials are created automatically and are associated with specific endpoints. Their names begin with '##'.

You can verify information about credentials by using the `sys.credentials` security catalog view, which returns one row for each credential in the server. The information contained in each row is listed in Table 11-7.

Table 11-7 Information in the `sys.credentials` Security Catalog View

Column Name	Data Type	Description
<code>credential_id</code>	int	ID of the credential. Is unique within the server.
<code>name</code>	sysname	Name of the credential. Is unique within the server.
<code>credential_identity</code>	nvar- char(4000)	Name of the identity to be used. This will generally be a Windows login. It need not be unique.
<code>create_date</code>	datetime	Time at which the credential was created.
<code>modify_date</code>	datetime	Time at which the credential was modified.

MORE INFO **Creating credentials**

For more information about creating credentials, search Books Online for “CREATE CREDENTIAL (Transact SQL)” or access [msdn2.microsoft.com/en-us/library/ms189522\(SQL.90\).aspx](https://msdn2.microsoft.com/en-us/library/ms189522(SQL.90).aspx).

Verifying Permissions to Roles and Accounts

You can use the Object Explorer component of SQL Server Management Studio to view the explicit permissions to a securable granted to a user account or database role, and to alter these permissions if they have been incorrectly configured. You can use the same tool to verify the effective permissions that result from the combination of inherited and explicit permissions.

Object Explorer User or Role Properties (Securables Page) You can use the Securables page in either the User or Role Properties dialog box to view or set explicit permissions on database securables for a user, database role, or application role. This page shows only permissions that are explicitly granted or denied to a user or role. The principal may inherit additional permissions through membership in a group or role. The sum of all the explicit and inherited (implicit) permissions of the principal

constitutes its effective permissions. From this page, you can access the following items:

- **User Name Or Role Name** The name of the principal (user or role) whose permissions are described.
- **Securables** The securables on which specific permissions have been granted or denied to this principal.
- **Schema** The schema that owns the object.
- **Name** The name of the securable.
- **Type** The type of the securable.
- **Add** Adds an object to the Securables grid.
- **Remove** Removes a securable.
- **Explicit permissions for <the selected securable>** Lists the permissions that can be granted or denied on the selected securable.
- **Grant** Select this option to grant a permission to the user or role. Deselect this option to revoke a permission.
- **With Grant** This reflects the state of the WITH GRANT option for the listed permission. The With Grant check box is read-only. To apply this permission, use the GRANT (Transact-SQL) statement.
- **Deny** Select this check box to deny this permission to the user or role. Deselect it to revoke this permission.

Object Explorer Effective Permissions Dialog Box The Effective Permissions dialog box in Object Explorer shows the result of combining explicit permissions with permissions inherited from groups and roles. The effective permissions that a principal has on a securable are the result of the explicit permissions defined for that principal on that securable, the permissions defined on the parents of the securable, and the permissions that the principal inherits through role or group membership. The Effective Permissions dialog box is read-only, and it displays the following information:

- **Principal** The name of the principal that possesses the permissions listed in the effective permissions grid.
- **Securable** The name of the securable.
- **Permission** The type of permission.
- **Column** Permission for a specific column of a table, view, or table-valued function. This is blank if the permission does not relate to a single column.

Deleting or Disabling Unused Accounts

A domain administrator (or an account operator) will delete Windows accounts or disable them if they are not currently used but might be required in the future. However, the DBA deletes or disables user accounts in a database that no longer require access to that database, and the DBA also deletes databases or database objects that are no longer required. Where SQL Server rather than Windows integrated authentication is used, the DBA can also deny user accounts access to a server. None of the DBA actions deletes or disables the actual Windows account.

Selecting the Status page in the Login Properties dialog box for a login lets you enable or disable a login, and grant or deny permission for the login to connect to a database. You can use the Delete Objects dialog box in Object Explorer to delete a database or database object. To delete multiple objects in the Delete Objects dialog box, you should press F7 to open the Summary Page, select multiple items, and then click Delete.

NOTE DROP DATABASE

You can also delete a database by using the DROP DATABASE command in Transact-SQL.

The Delete Objects dialog box offers the following additional controls:

- **Show Dependencies** You can click Show Dependencies to display both the objects that are dependent on the currently selected object and objects on which the current object is dependent (upward and downward dependency). The information displayed in the Show Dependencies dialog box is read-only.

NOTE Show Dependencies does not always appear.

The Show Dependencies control does not appear for all types of database objects. To view dependencies when Show Dependencies is not available, you can right-click the object in Object Explorer, and then choose View Dependencies.

- **Delete Backup And Restore History Information For Databases** This check box deletes the backup and restore history for the subject database from the msdb database. It appears only when a database is deleted.
- **Close Existing Connections** This check box terminates connections to the subject database. It appears only when a database is deleted.

PRACTICE Using Object Explorer

Object Explorer is a component of SQL Server Management Studio. It connects to database engine instances, Analysis Services, Integration Services, Reporting Services, and SQL Server Mobile. Object Explorer provides a view of all the objects in the server and presents a user interface to manage them. The capabilities of Object Explorer vary slightly depending on the type of server to which it is connected, but they generally include the development features for databases and management features for all server types.

NOTE Filtering the list of objects in Object Explorer

When a folder contains a large number of objects, it can be difficult to find the object you are looking for. You can use the filter feature of Object Explorer to reduce the list to a smaller size. Select the folder that you want to filter, and then click the Filter button to open the Object Explorer Filter Settings dialog box. You can filter the list by name, creation date, and sometimes schema, and you can specify additional filtering operators such as Starts With, Contains, and Between.

► Practice 1: Connecting Object Explorer and Registering a Server

In the following practice, you connect Object Explorer to a server and a database on that server. The practice assumes that Object Explorer is visible when you open SSMS. If you cannot see Object Explorer, click the Object Explorer control in SSMS, as shown in Figure 11-5.

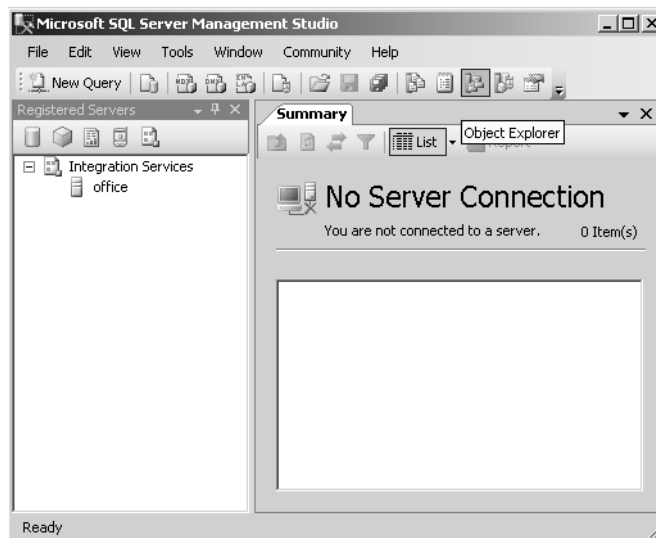


Figure 11-5 The Object Explorer control in SQL Server Management Studio.

1. Log in to your domain at your member server by using an account that has the sysadmin server role on your member server. If you are using virtual machine software, log in to your domain and connect to your member server.
2. On the All Programs (or Programs) menu, select Microsoft SQL Server 2005, and then select SQL Server Management Studio.
3. In the Connect To Server dialog box, specify Database Engine as the server type. Specify the name of your member server as the server name. (This might differ from the one shown, depending upon your setup.) Specify Windows Authentication. The Connect To Server dialog box is shown in Figure 11-6.



Figure 11-6 The Connect To Server dialog box.

4. Select Options. On the Connection Properties tab, specify TCP/IP as the network protocol. On the Connect To Database drop-down menu, select Browse Server and select the AdventureWorks user database.
5. Observe but do not alter the other settings on this tab. Note that the default execution timeout of 0 indicates that execution will never time out.
6. Click Connect.
7. In Object Explorer, right-click the server name, and then choose Register, as shown in Figure 11-7.
8. Specify a user-friendly server name and type a description, as shown in Figure 11-8.
9. Click New Group. Add a group name and description. Click Save.
10. Close SSMS.

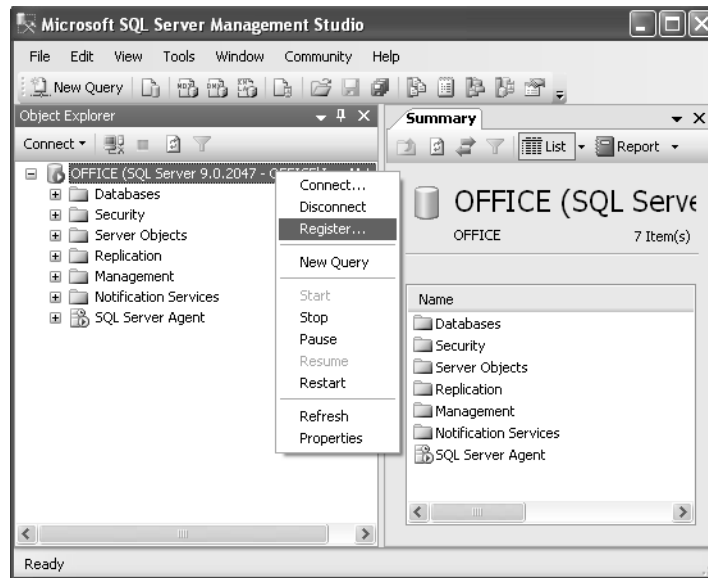


Figure 11-7 Registering a server from Object Explorer.

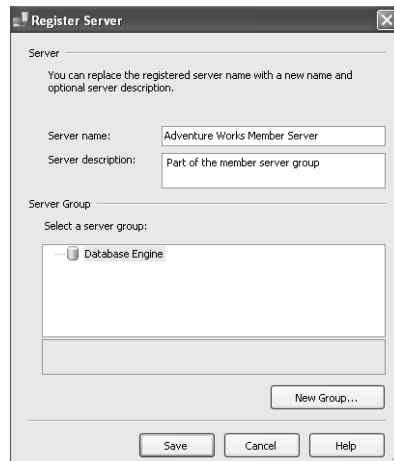


Figure 11-8 Registering a server with a user-friendly name.

► Practice 2: Using Object Explorer to Verify Roles and Logins

In the following practice, you use Object Explorer to verify (and if necessary configure) server role membership and reconfigure login properties.

1. Log in to your domain at your member server by using an account that has the sysadmin server role on your member server. If you are using virtual machine software, log in to your domain and connect to your member server.

2. On the All Programs (or Programs) menu, select Microsoft SQL Server 2005, and then select SQL Server Management Studio.
3. In the Connect To Server dialog box, specify Database Engine as the server type. Specify the name of your member server as the server name. (This might differ from the one shown, depending upon your setup.) Specify Windows Authentication.
4. Click Connect.
5. In Object Explorer, expand Security, expand Server Roles, and double-click sysadmin. The members of the sysadmin role are listed, as shown in Figure 11-9. If you have not already done so, it is a good idea to remove sa from this role. Click OK to close the Server Role Properties – Sysadmin dialog box.

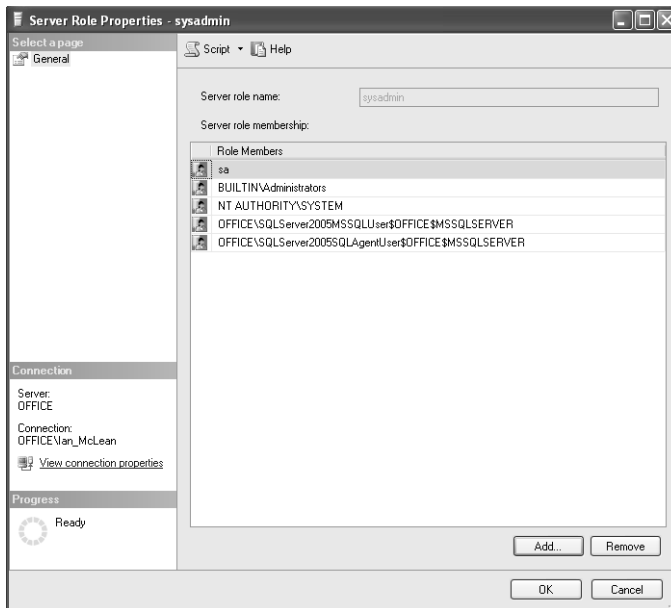


Figure 11-9 Members of the sysadmin role.

6. In Object Explorer, expand Logins and double-click a login. On the General page of the Login Properties dialog box, you can specify the default database and default language for this login. If SQL authentication is used, you can also specify a new password and specify whether to enforce password policy and password expiration. Do not make any changes at this time.
7. Select the Server Roles page. You can add or remove server roles from an individual login or login group, as shown in Figure 11-10.
8. Select the User Mapping page. This page lets you select a database and add the login to the database roles in that database.

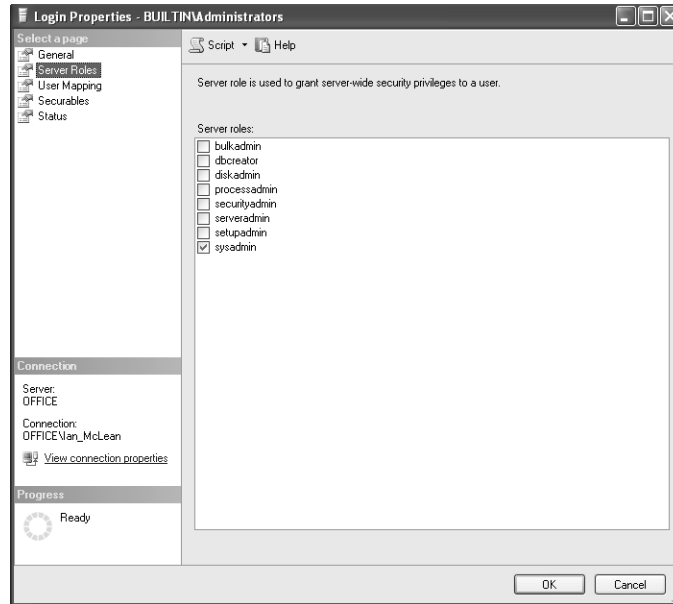


Figure 11-10 Adding a login to server roles.

9. Select the Securables page. This page lets you select a securable and grant the login explicit permissions to that securable as shown in Figure 11-11. You can use the controls on this screen to add or remove securables from the list and to view the effective permissions on a securable granted to the login user or group account.

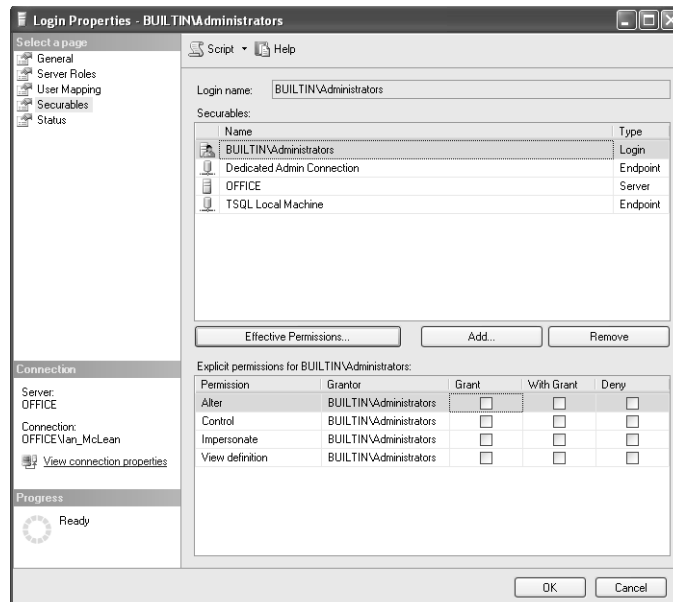


Figure 11-11 Granting a login explicit permissions to a securable.

10. Select the Status page. This page lets you enable or disable the login, and grant or deny permission for the login to connect to the database.
11. Click OK to close the Login Properties dialog box.
12. Close SSMS.

NOTE Opening a connected Query Editor

When Object Explorer is connected to a server, you can open a new Code Editor window by right-clicking the server name, and then choosing New Query. To open a Code Editor window using a particular database, right-click the database name and then choose New Query. When opening a new query for an Analysis Services server, you can select DMX, MDX, or XMLA queries.

► Practice 3: Using Object Explorer to View SQL Server and Windows Error Logs

In the following practice, you use Object Explorer to view Database Mail, SQL Server Agent, SQL Server, and Windows error logs.

1. Log in to your domain at your member server by using an account that has the sysadmin server role on your member server. If you are using virtual machine software, log in to your domain and connect to your member server.
2. On the All Programs (or Programs) menu, select Microsoft SQL Server 2005, and then select SQL Server Management Studio.
3. In the Connect To Server dialog box, specify Database Engine as the server type. Specify the name of your member server as the server name. (This might differ from the one shown, depending upon your setup.) Specify Windows Authentication.
4. Click Connect.
5. In the Object Explorer pane, expand SQL Server Agent and then expand Error Logs. Double-click the current error log.
6. In the Select Logs pane of Log File Viewer, expand Database Mail, SQL Agent, SQL Server, and Windows NT.
7. Select the Database Mail log. Unless you previously enabled database mail, this log should be empty.
8. Select the current SQL Agent log, as shown in Figure 11-12. You can filter logs by connection, start and end date, source, or specified text content.
9. Select the current SQL Server log, and clear the check box for the SQL Agent log. Typically, the SQL Server log has similar entries to the SQL Agent log.

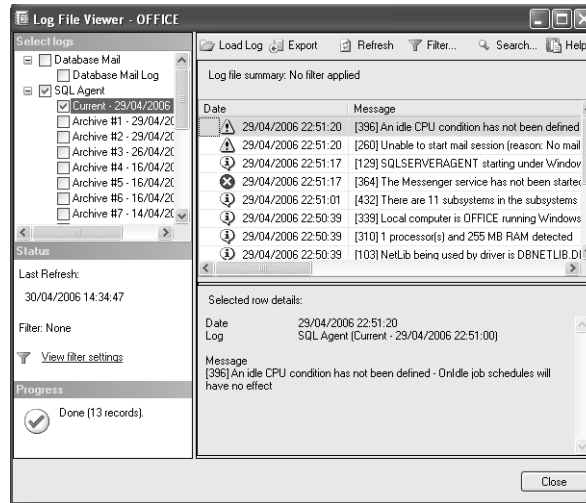


Figure 11-12 The current SQL Server Agent log.

10. Select the Windows NT Applications, Security, and Systems logs in turn. Figure 11-13 shows the Windows NT System log.

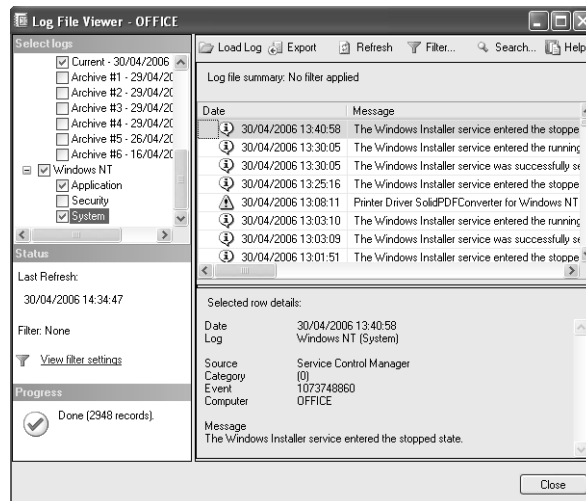


Figure 11-13 The Windows System log.

11. Close the Log File Viewer and then close SSMS.

NOTE Refreshing Object Explorer

Folders in the Object Explorer tree do not automatically refresh their list of contents. To refresh the list of objects within a folder, right-click the folder and then click Refresh.

Lesson Summary

- User accounts that access databases on SQL Server require strong password policies.
- The RSoP tool assists in determining whether password and other security policies have been correctly configured.
- You can use Transact-SQL catalog views, in particular `sys.sql_logins`, and the `LOGINPROPERTY` Transact-SQL function to verify login authentication.
- SQL Server 2005 supports both Windows integrated and Mixed authentication modes. The Mixed authentication mode allows both Windows and SQL Server authentication. You should use Windows integrated authentication where possible.
- You can use the `sp_helprolemember` stored procedure to list and verify the accounts that are members of a database role.
- You can use the `sp_helpuser` stored procedure to list the users in the current database and to obtain information about database roles and database-level principals.
- You can verify information about credentials by using the `sys.credentials` security catalog view.
- You can use the Object Explorer component of SQL Server Management Studio to view the explicit permissions to a securable granted to a user account or database role. You can use the same tool to verify the effective permissions that result from the combination of inherited and explicit permissions.
- Selecting Status in the Login Properties dialog box in Object Explorer lets you enable or disable a login, and grant or deny permission for the login to connect to a database. You can use the Delete Objects dialog box in Object Explorer to delete a database or database object.

Lesson Review

You can use the following questions to test your knowledge of the information in Lesson 2, “Maintaining a User-Level Security Strategy.” The questions are also available on the companion CD if you prefer to review them in electronic form.

NOTE Answers

Answers to these questions and explanations of why each answer choice is correct or incorrect are located in the “Answers” section at the end of the book.

1. You are a domain administrator, and you also have database administration responsibilities. You have altered the domain structure, reconfigured inheritance, and changed the Windows permissions on several user and group accounts. You want to check your configuration. What is the first step you should take? (Choose all that apply. Each choice represents a complete answer.)
 - A. Open Active Directory Users And Computers, and start the Resultant Set Of Policy Wizard.
 - B. Open SQL Server Management Studio. In Object Explorer, navigate to the effective permissions dialog box.
 - C. Open Active Directory Sites And Services, and start the Resultant Set Of Policy Wizard.
 - D. Open SQL Server Management Studio. In Object Explorer, expand Security and navigate to the Login Properties dialog box.
2. Which Transact-SQL features can you use to verify SQL Server login authentication? (Choose all that apply.)
 - A. The *sp_helpsrvrolemember* stored procedure
 - B. The LOGINPROPERTY Transact-SQL function
 - C. The sys.sql_logins Transact-SQL server-level security catalog view
 - D. The sys.server_permissions Transact-SQL server-level security catalog view
3. You are a DBA with overall responsibility for SQL Server and database management in your organization. You want your assistant, Kim Akers, to be able to manage logins and login properties. She should be able to grant, deny, and revoke server-level and database-level permissions, and reset passwords for SQL Server logins. To which fixed server role should you add Kim's account?
 - A. processadmin
 - B. serveradmin
 - C. sysadmin
 - D. securityadmin

Chapter Review

To further practice and reinforce the skills you learned in this chapter, you can complete the following tasks:

- Review the chapter summary.
- Review the list of key terms introduced in this chapter.
- Complete the case scenarios. These scenarios set up real-world situations involving the topics of this chapter and ask you to create a solution.
- Complete the suggested practices.
- Take a practice test.

Chapter Summary

- Service accounts should meet the same security criteria as ordinary user accounts, although typically they have longer and more complex passwords. You should apply the principle of least privilege to both service accounts and ordinary user accounts. You can use SQL Server Management Studio to configure SQL Server auditing.
- Sensitive data should be encrypted. Encryption/decryption methods used by SQL Server are: password, symmetric key, asymmetric key, and certificate.
- You should test service packs and security updates on a preproduction network before they are installed on a production network.
- A SQL Server 2005 server should present the smallest possible surface area for attack.
- The RSoP tool assists in determining whether password and other security policies have been correctly configured.
- SQL Server 2005 supports both Windows integrated and Mixed authentication modes. You can use Transact-SQL catalog views, in particular `sys.sql_logins`, and the `LOGINPROPERTY` Transact-SQL function to verify login authentication.
- You can use stored procedures to list and verify the accounts that are members of a database role, list the users in the current database, obtain information about database roles and database-level principals, and verify information about credentials.
- You can use the Object Explorer component of SQL Server Management Studio to verify the permissions to a securable granted to a user account or database role, to enable or disable a login, and to grant or deny permission for the login to connect to a database.

Key Terms

Do you know what these key terms mean? You can check your answers by looking up the terms in the glossary at the end of the book.

- certificate authority
- credentials
- cryptography
- endpoint
- metadata
- privileges
- securable
- security context
- security principal

Case Scenarios

In the following case scenarios, you will apply what you've learned about maintaining server-level and user-level security strategies. You can find answers to these questions in the “Answers” section at the end of this book.

Case Scenario 1: Configuring Security on SQL Server 2005 Member Servers

You are a senior DBA working for Contoso, Ltd. You do not have any permissions to administer the company's Active Directory domain. You are responsible for six SQL Server 2005 SP1 member servers and the databases they contain, and your DBA user account has been added to the sysadmin role on these servers. The written company policy specifies the following:

- Domain users should be able to access SQL Servers and run queries against any databases to which they are allowed access without needing to supply additional credentials.
- Database administrators should be able to manage the SQL Server member servers, but they should not have privileges on any other part of the company network.
- Service packs and security updates are obtained from the Microsoft Update Web site and distributed by two WSUS servers. This operation is managed by the domain administrators team, but you need to advise them on any updates that affect the SQL Server servers.

Answer the following questions:

1. What type of authentication should you specify for the SQL Server member servers? What possible problems can you foresee with password policy settings?
2. What should you ask the domain administration team to do to enable you and your team to administer the SQL Server member servers?
3. Currently Contoso has no preproduction network. Management has asked you to justify the expense of such a network. How should you reply?

Case Scenario 2: Adding Your Team Members' User Accounts to Database Roles

You are a senior DBA working for Northwind Traders. Your database administration team consists of David Alexander, Don Hall, Carol Philips, and Matt Berg. You want your team members to perform the following tasks on the Northwind Traders Production database:

- All members of your team should be able to read all data from all user tables and add, delete, or change data in all user tables.
- David and Don should be able to add or remove access for Windows logins, Windows groups, and SQL Server logins.
- Don and Carol should be able to back up the database.
- Matt should be able to modify role membership and manage permissions. He should not, however, have DBO rights to the database or be able to add members to the db_owner database role.

Answer the following questions:

1. To what database roles should you add your team members' user accounts?
2. You decide that Don should also be able to run any DDL commands in the database. To what additional database role should you add his account?

Suggested Practices

To help you successfully master the exam objectives presented in this chapter, complete the following practice tasks:

Maintain a Server-Level Security Strategy

You should complete this practice on your test network before configuring the surface area on your production servers. The results you get will vary because your test network is probably set up differently from your production servers.

- **Practice 1: Configure the surface area.** Configure the surface area for the master system database on your member server. Document any differences from the practice in Lesson 1 in which you specified the AdventureWorks database.

Maintain a User-Level Security Strategy

You should complete these practices on your test network before you use the same techniques to obtain security information on your production servers. You might not have permission to run the RSoP wizard on your production server, but you can ask a domain administrator to run the tool and let you see the results.

- **Practice 1: Run the RSoP wizard.** Run the RSoP wizard to discover the effects changes to Windows permissions have on your test network. Change server and database role membership, and run the wizard again to discover whether your actions affect Windows permissions.
- **Practice 2: Use catalog views and Transact-SQL functions.** Access Books Online to find out more about catalog views and Transact-SQL functions. Use Transact-SQL `sys.sql_logins` and `LOGINPROPERTY` to verify login authentication.
- **Practice 3: Use stored procedures.** Access Books Online to find out more about stored procedures. Use the relevant stored procedures to obtain information about (for example) database users, role membership, and database-level principals.
- **Practice 4: Become familiar with Object Explorer.** Object Explorer gives you access to a great deal of information, and it is possible to cover only a few of its screens and dialog boxes in this chapter. Only extensive hands-on practice will familiarize you with this tool.

Take a Practice Test

The practice tests on this book's companion CD offer many options. For example, you can test yourself on just one exam objective, or you can test yourself on all the 70-444 certification exam content. You can set up the test so that it closely simulates the experience of taking a certification exam, or you can set it up in study mode so that you can look at the correct answers and explanations after you answer each question.

MORE INFO Practice tests

For details about all the practice test options available, see the "How to Use the Practice Tests" section in this book's Introduction.

Chapter 12

Detecting and Responding to Attacks

The more valuable the data stored on the Microsoft SQL Server 2005 computer that you manage, the more likely it is that the server will come under some form of attack. Attacks are usually external—a hacker attempting to gain access to credit-card data or someone showing off to his or her online friends. Internal attacks can be more sinister because the attacker is someone known and theoretically trusted by the organization. This chapter discusses the nature of both kinds of attacks and the preventative measures that a database administrator (DBA) can take.

Exam objectives in this chapter:

- Perform a security audit of the existing security based on the security plan.
- Analyze the physical server security.
- Compare the existing security infrastructure to business and regulatory requirements.
- Identify variations from the security design.
- Prepare for and respond to threats and attacks.
- Prepare for and respond to SQL Server injection attacks.
- Prepare for and respond to denial-of-service attacks that are specific to SQL Server.
- Prepare for and respond to virus and worm attacks that are specific to SQL Server.
- Prepare for and respond to internal attacks that are specific to SQL Server.

Lessons in this chapter:

- Lesson 1: Auditing the Existing Infrastructure. 649
- Lesson 2: Protecting Against Threats and Attacks 660

Before You Begin

To complete the lessons in this chapter, you must have completed the following tasks:

- Configured a Microsoft Windows Server 2003 R2 computer with Microsoft SQL Server 2005 Enterprise Edition SP1 as detailed in the Appendix.
- Installed an updated copy of the AdventureWorks sample database as detailed in the Appendix.

No additional configuration is required for this chapter.

Lesson 1: Auditing the Existing Infrastructure

The first step to take in protecting a SQL Server 2005 installation is to closely examine the physical environment that will house this important piece of organizational infrastructure. This step boils down to answering the question, “How much effort would it take for anyone in the office to place his or her hand on the server?” Auditing the existing infrastructure also means knowing exactly which service packs, updates, and hotfixes have been applied to a computer, and which service packs, updates, and hotfixes have yet to be applied because they have not yet passed through the organization’s update management cycle.

After this lesson, you will be able to:

- Analyze physical server security.
- Compare the existing security infrastructure to business and regulatory requirements.
- Identify variations from the security design.

Estimated lesson time: 30 minutes

NOTE Defining “attacker”

For the purposes of this chapter, an attacker is a person (or more than one person) who wants to gain unauthorized access to the database. Attackers can be people from outside or inside the organization.

Analyzing Physical Server Security

If an attacker has physical access to a server, he has access to all that server’s data. If an attacker is standing in front of the server, the server is as good as compromised because the attacker can then do any of the following:

- Boot up the server using a bootable CD-ROM, and copy all the server data to a removable drive, bypassing logon security.
- Install a keylogger to get access to Administrator passwords.
- Use a screwdriver to open the server and leave the building with the hard disk drives.

Because servers are critical components in a business’s infrastructure, it is important that they be housed in a safe location. For example, you should not keep servers on

the cubicle desk next to the system administrator's workstation, but rather in a locked room that only a select number of authorized personnel have access to.

The room in which you store the servers should have the following properties:

- **The door to the server room must have an electronic lock.** Standard keys are easily copied. This lock should use two-factor authentication involving elements such as fingerprints, PINs or passwords, Radio Frequency Identification (RFID) tags, or swipe cards. The door should automatically lock when several failed authentications occur.

NOTE Two-factor authentication

Two-factor authentication requires two independent ways to establish identity. This can be a mix of smart card and password, fingerprint and password, or smart card and fingerprint. Both elements must be present for authentication to be possible.

- **Entry and exit from the server room should be logged.** If the power fails, the door should remain locked from the outside. For safety reasons, people inside should be able to exit. In such a situation, their entry is already logged.
- **A list should exist of all people who have physical access to the server.** This list should be regularly reviewed. People who no longer require access should have their access disabled. People who are on the list but, according to the log, never actually access the room should be removed from the list. The cleaners who clean the building at night should not have access to the server room to vacuum the floor. Unless the database administrators can organize a way to supervise the cleaners, the DBAs will need to use a vacuum cleaner!
- **Video surveillance equipment should be installed.** On a regular basis, the door log should be compared to the video log to check whether unauthorized access is occurring. For example, if the log says that Ian entered the room at 9:03 A.M. and exited at 9:17 A.M., yet the video log shows only Orin in the room at 9:10 A.M., something is clearly amiss. Comparing these two separate records verifies that both security devices are working.
- **The ceilings of the server room need to be waterproof, have an appropriate fire rating, and be reinforced to keep attackers from climbing through them into the server room.** Two-factor authentication is pointless if someone can climb through the vents or remove some ceiling tiles to get into the server room. If the sprinklers in the rooms on the floor above the server room are activated, under no circumstances should the server room become flooded.
- **The server room should use a halon fire suppression system.** These systems dump gas, rather than water, into the server room when a fire is detected. Halon is a gas

that stops the spread of fire by chemically disrupting combustion. Halon leaves no residue, is rated safe for human exposure, and is extremely effective at extinguishing flammable liquid and electrical fires. Halon gas should be used because traditional sprinkler systems will most likely destroy any electronic equipment.

- **Ensure that backup tapes are stored securely.** Backup tapes should be stored in a fireproof safe rather than on the shelf near the system administrator's desk. If possible, have tapes rotated to a secure offsite facility so that if the building is destroyed by fire or natural disaster, the organization's data can be restored.
- **Regularly inspect servers for the presence of unusual attached devices such as keyloggers.**

Exam Tip In the real world and on the exam, there is no single list of steps that need to be taken to ensure a server is physically secure. Remember to separate physical security methods (such as locked doors) from logical methods (such as disabling protocols) when answering a question.

In many organizations, implementing all these aspects of physical security isn't economically justifiable. Cost-benefit analysis is a fundamental part of security, and what an organization spends on protecting an investment should never be worth more than the investment itself.

Real World

Orin Thomas

The more secure something is, the more inconvenient and more expensive it is. The more inconvenient and expensive it is, the more likely it is that people are going to complain to you about it. Unless you approach the discussion in the right way, you might find it difficult to argue that a server room that has been protected only by a lock and key for the last 10 years suddenly needs a smart card reader and a keypad device. A big part of implementing any security plan is explaining decisions to those who will be influenced by the plan's practical implementation. When people understand why you are instituting strict security policies, they are more likely to accept those policies rather than feel that something arbitrary has been forced upon them.

Securing Administrator Workstations

One issue that many organizations do not consider is the security of their administrators' workstations. Just as servers need to be secured, the workstations of database administrators need to be similarly secured. If an attacker attaches a keylogger to a

database administrator's workstation, the attacker will soon have all the database administrator's login and password information. Many organizations make the mistake of locking up the servers but placing the administrators who manage those servers out in cubicles, where people can shoulder-surf passwords or attach keylogger hardware when the administrator is attending meetings or out to lunch. A good step is to place the sysadmin team in an area that can be accessed only via two-factor logged authentication. Administrators should also be trained to regularly inspect their workstations for unusual devices that might be used to record their actions.

CAUTION Keyloggers

Keyloggers are small devices that are attached between the end of a keyboard cable and the keyboard port on the computer. They come in USB and PS2 varieties. They record each keystroke entered. The attacker attaches the keylogger, hopes that the keylogger is not noticed, and removes it at a later stage. Unless you check the back of your computer each time you come back to it, you won't notice whether a keylogger has been installed.

SQL Server Security Considerations

Securing a SQL Server instance requires more than a locked room. You should also address certain network infrastructure considerations, as detailed in the following list:

1. Ensure that the computer on which SQL Server is installed is never directly exposed to the Internet. You should install SQL Server only on the internal network. SQL Server should not be installed on a screened subnet. If a Web application is installed on an Internet Information Services (IIS) server on the screened subnet, you should configure the internal firewall to allow communication between these two hosts only. Figure 12-1 provides a suggested network layout.
2. In a domain environment where all clients run Microsoft Windows XP or Vista and all servers run Microsoft Windows Server 2003 or later, disable NTLM authentication.
3. Where possible, do not install SQL Server 2005 on a domain controller.
4. Configure SQL Server services to run under separate Windows accounts.
5. Disable NetBIOS and Server Message Block (SMB). NetBIOS uses UDP ports 137 and 138 and TCP port 139. SMB uses TCP ports 139 and 445.

MORE INFO Disabling NetBIOS and SMB

The following MSDN articles discuss how to disable NetBIOS and SMB: msdn2.microsoft.com/en-us/library/ms143696.aspx and msdn2.microsoft.com/en-us/library/ms143455.aspx.

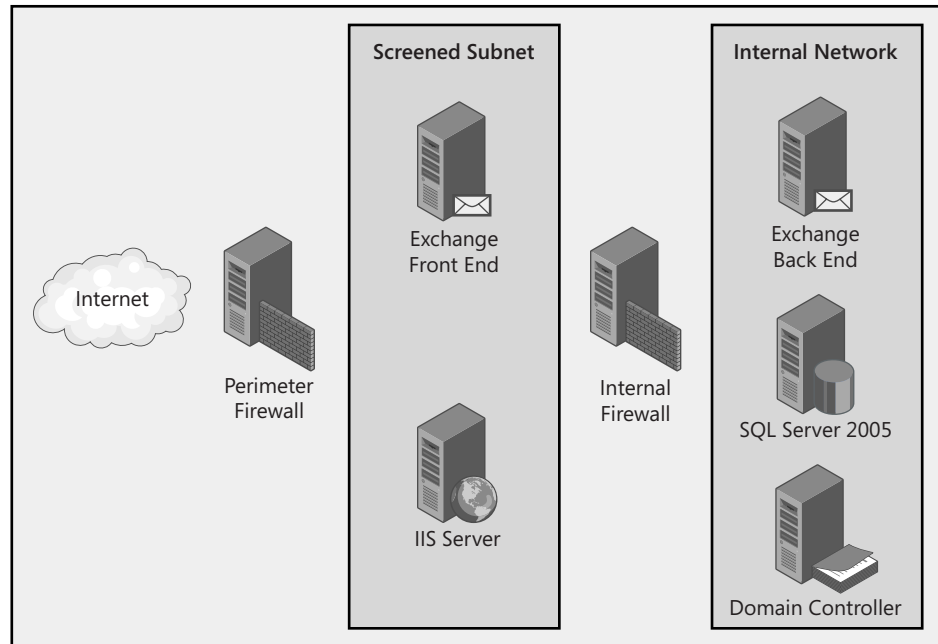


Figure 12-1 Protecting SQL Servers from the Internet by placing them on the internal LAN.

6. Consider restricting the SQL Server so that it will communicate only with hosts it can negotiate an IPSec connection with.
7. Configure Web applications that interact with SQL Server to do so using an encrypted Secure Sockets Layer (SSL) connection.

Quick Check

1. Name at least two properties that a server room door lock should have.
2. Why should administrators regularly inspect their workstations?

Quick Check Answer

1. The properties a server room door lock should have include the following: two-factor authentication, logging access, locks from the outside in the event of a power failure, and an exit that allows people who are already inside the room to leave during a power failure.
2. Administrators need to regularly inspect their workstations to ensure that keylogger devices haven't been attached.

Security Configuration And Analysis

The Security Configuration And Analysis snap-in is a tool you use for analyzing and configuring local system security. Regular analysis allows administrators to ensure that an adequate level of security exists on the servers they are responsible for managing. The security levels can be fine-tuned and used to detect any security flaws that become apparent on the computer over time. Security configuration analysis presents recommendations by using visual flags or remarks to highlight areas where current settings do not match the proposed level of security, as shown in Figure 12-2. You can also use the Security Configuration And Analysis snap-in to configure the security of the local computer by applying security templates directly to the local computer.

Policy	Database Setting	Computer Setting
Audit account logon events	Success, Failure	Success
Audit account management	Success, Failure	Success
Audit directory service access	Not Defined	Success
Audit logon events	Success, Failure	Success
Audit object access	Success, Failure	No auditing
Audit policy change	Success, Failure	Success
Audit privilege use	Success, Failure	No auditing
Audit process tracking	No auditing	No auditing
Audit system events	Success, Failure	Success

Figure 12-2 A comparison between a server's settings and those in a high-security template.

The predefined security templates that are provided with Security Configuration And Analysis provide a starting point for creating security policies. The predefined templates are designed for general computer roles such as Domain Controller and Workstation. None of the default security templates are specifically appropriate for a SQL Server 2005 computer. Database administrators need to create an individualized template that is appropriate for their particular deployment. After you create the template, you can analyze other SQL Servers in the organization using the template. You can also apply the template to other SQL Servers in the organization as a method of standardizing their security configuration. As mentioned earlier, you can also use templates on a regular basis to check that security settings have not deviated over time.

Using Secedit to Perform Security Configuration and Analysis

Secedit is a command-line tool that can perform the same tasks that the Security Configuration And Analysis tool can. One advantage of secedit is that you can include it in batch files. Database administrators can use a combination of batch file and scheduled tasks to ensure that a particular security template is reapplied to a server on an ongoing basis.

To perform an analysis, with a security database named `secdbname.sdb` and using security template `Templatename`, enter the following command at the command prompt:

```
secedit /analyze /db secdbname.sdb /cfg Templatename
```

Secedit creates the security database you specify (`secdbname.sdb` in our example) when you run the command. Secedit uses this database to store the results of its analysis of your computer's security settings.

To apply a security template named `Templatename` to a computer, using the security database named `secdbname.sdb`, enter the following command at the command prompt:

```
secedit /configure /db secdbname.sdb /cfg Templatename
```

Troubleshooting Security Configuration

If a computer is shut down suddenly, the security database can become corrupted. If the security database becomes corrupted, perform the following steps to repair it:

1. At the command prompt, run **esentutil /g** to check the integrity of the security database. The database is found at `%windir%\Security\Database\secedit.sdb`.
2. If the database is corrupt, change to the `%windir%\Security` folder and run **esentutil /r**.
3. If this does not work, run **esentutil /p** against `%windir%\Security\Database\secedit.sdb`.
4. If none of the previous three steps work, delete all log files located in `%windir%\Security`.

MORE INFO Secedit

More information about secedit can be found at technet2.microsoft.com/WindowsServer/en/Library/b1007de8-a11a-4d88-9370-25e2445605871033.msp?mfr=true.

Using the MBSA Tool to Audit Security

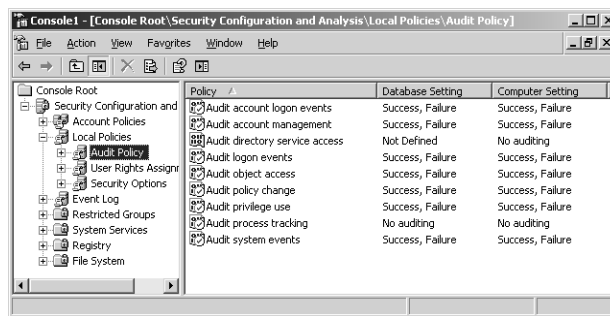
As described in Chapter 6, “Database Maintenance,” you can use the Microsoft Baseline Security Analyzer (MBSA) tool to scan a server to determine which security updates and service packs have yet to be installed. Although following update-testing best practices means that SQL Server 2005 computers in most organizations won’t have every update installed until they have gone through a rigorous evaluation process, you can still use the MBSA tool to analyze each server on a regular basis. Printouts of these reports can be stored for historical purposes. You can also use the MBSA tool printout as a roadmap to remind you which updates still need to go through the evaluation process and then be applied to either SQL Server 2005 or the underlying operating system.

PRACTICE Configuring Security Using Templates

In the following exercise, you configure the security on server Glasgow using the HISECWS security template. To complete this exercise, perform the following steps:

1. Log in to server Glasgow using the Administrator account.
2. From the Start menu, choose Run.
3. Enter **MMC** and click OK.
4. From the File menu, choose Add/Remove Snap-in.
5. In the Add/Remove Snap-in dialog box, click Add.
6. Select Security Configuration And Analysis, and click Add. Click Close. Click OK.
7. Right-click the Security Configuration And Analysis scope item.
8. Choose Open Database.
9. Type **SQL2005** and then click Open, which creates a new security database.
10. Select hisecws.inf and click Open.
11. Right-click the Security Configuration And Analysis scope item.
12. Choose Analyze Computer Now.
13. Click OK to accept the default error log path.
14. Expand the Security Configuration And Analysis scope item.
15. Expand the Local Policies folder.

16. Select Audit Policy. Examine the differences between the database settings and the computer settings.
17. Right-click the Security Configuration And Analysis scope item.
18. Choose Configure Computer Now.
19. Click OK to accept the default error log file path.
20. Right-click the Security Configuration And Analysis scope item.
21. Choose Analyze Computer Now.
22. Click OK to accept the default error log file path.
23. Expand the Local Policies folder.
24. Select Audit Policy. Re-examine the differences between the database settings and the computer settings. They should now be the same, as shown in Figure 12-3.



Policy	Database Setting	Computer Setting
Audit account logon events	Success, Failure	Success, Failure
Audit account management	Success, Failure	Success, Failure
Audit directory service access	Not Defined	No auditing
Audit logon events	Success, Failure	Success, Failure
Audit object access	Success, Failure	Success, Failure
Audit policy change	Success, Failure	Success, Failure
Audit privilege use	Success, Failure	Success, Failure
Audit process tracking	No auditing	No auditing
Audit system events	Success, Failure	Success, Failure

Figure 12-3 After you apply a security policy, a server's security settings match that of the policy.

Lesson Summary

- Physical server security is about ensuring that attackers are unable to physically access a server.
- Server room doors should use two-factor authentication, record a log, and lock if several authentication failures occur in a row.
- You should keep a list of all people who require access to the server room. People who are on the list but, according to the log, never actually enter the room, should be removed from the list.
- The server room should have video surveillance. The video log should be compared to the log from the door on a regular basis.

- Halon fire suppression rather than water-based fire suppression should be installed so that if there is a fire in the server room, the servers won't be damaged by water while the fire is being extinguished.
- Backup tapes should be stored securely in a fireproof safe. If possible, they should be rotated to an offsite location.
- Servers and administrator workstations should be regularly inspected for unusual hardware devices, such as keyloggers, that might be used to snoop for passwords.
- SQL Server should never be directly exposed to the Internet. It should always be installed on a network that is protected from the Internet by at least one firewall.
- Use security templates to set and preserve security configurations.

Lesson Review

You can use the following questions to test your knowledge of the information in Lesson 1, "Auditing the Existing Infrastructure." The questions are also available on the companion CD if you prefer to review them in electronic form.

NOTE Answers

Answers to these questions and explanations of why each answer choice is correct or incorrect are located in the "Answers" section at the end of the book.

1. Which of the following should you use to secure the door of a server room? Each answer represents an individual choice.
 - A. A video surveillance camera
 - B. A smartcard reader
 - C. A fingerprint reader
 - D. An RFID tag reader and a PIN number keypad
2. The door to your organization's server room has been configured with a fingerprint reader and a magnetic swipe card reader. Inside the server room, a video surveillance camera has been installed. You suspect that either the fingerprint reader or the magnetic swipe card reader is not functioning properly because it is logging people in the audit log with the wrong identity. Which of the following procedures would allow you to determine which piece of equipment is problematic?

- A. Check the audit log from the fingerprint reader and the magnetic swipe card reader against the video footage.
 - B. Check the audit log from the fingerprint reader against the video footage.
 - C. Check the audit log from the magnetic swipe card reader against the video footage.
 - D. Check the audit log from the fingerprint reader against the audit log from the magnetic swipe card reader.
3. You are in the process of creating a strategy to manage your organization's backup tapes. Which of the following elements should you include in your strategy? (Choose all that apply.)
- A. Rotation to a 24-hour, 7-days-a-week offsite facility
 - B. Fireproof safe
 - C. Filing cabinet
 - D. Bank safe deposit box
4. Which of the following security precautions should you take when installing SQL Server 2005? (Choose all that apply.)
- A. Don't install SQL Server 2005 on a domain controller.
 - B. Don't install SQL Server 2005 on a member server.
 - C. Configure all SQL Server services to run under the same Windows accounts.
 - D. Configure all SQL Server services to run under separate Windows accounts.

Lesson 2: Protecting Against Threats and Attacks

All database administrators need to take precautions to protect the database servers that they manage. In this lesson, the four most common types of attacks and the precautions that can be taken against these attacks are explored. Although taking precautions will not protect a database from all attacks, it will minimize the chance that attacks will be successful.

After this lesson, you will be able to:

- Prepare for and respond to SQL Server injection attacks.
- Prepare for and respond to denial of service attacks that are specific to SQL Server.
- Prepare for and respond to virus and worm attacks that are specific to SQL Server.
- Prepare for and respond to internal attacks that are specific to SQL Server.

Estimated lesson time: 30 minutes

Preparing for and Responding to SQL Server Injection Attacks

SQL Server injection attacks involve an attacker executing unauthorized SQL statements against a database by taking advantage of Web applications that fail to properly check user input. The attacker enters a malformed statement in a Web application text input box that, when processed by the Web application, allows the attacker to gain access to the back-end database. There are two types of SQL injection attack:

- Direct insertion of SQL code into user-input variables that are linked with Transact-SQL commands and then executed.
- Injection of malicious code into strings that are to be stored in tables or as metadata. When these stored strings are linked into a dynamic Transact-SQL command, malicious code is executed.

The following script and example show how an injection attack can occur. The script creates a Transact-SQL query by combining hard-coded strings with string input from a user:

```
var SampleSite;
SampleSite = Request.form ("SampleSite");
var sql = "select * from SampleTable where SampleSite = '" + SampleSite + "'";
```

In this example, the user is prompted to enter the name of the site where a particular sample was taken from. If the user enters **Kakadu**, the Transact-SQL query that the script generates will be as follows:

```
SELECT * FROM SampleTable where SampleSite = 'Kakadu'
```


If the user was somewhat more nefarious, she could enter the following text in the input box:

```
Kakadu'; DROP TABLE SampleTable--
```

The script would then generate the following Transact-SQL query:

```
SELECT * FROM SampleTable WHERE SampleSite = 'Kakadu';DROP TABLE SampleTable--'
```

Semicolons are used to denote the end of a query, and double hyphens are used to denote that the rest of the current line is a comment. If this Transact-SQL code is processed by SQL Server, the following will happen:

- All records in SampleTable where the SampleSite column is Kakadu will be selected.
- An attempt will be made to drop SampleTable.

It is not possible to detect tampering programmatically as long as the injected Transact-SQL code is syntactically correct. This means that database administrators must ensure that the Web application code screens all user input that is used to construct Transact-SQL statements that will be processed by SQL Server 2005.

Validating Input

User input can be validated by checking type, length, format, and range, as shown in Figure 12-4.

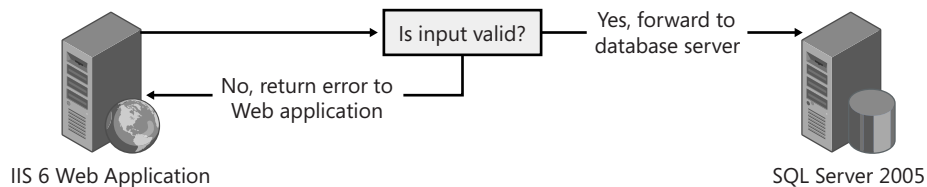


Figure 12-4 Input should be validated before it is passed to SQL Server.

The following should be considered when designing an application that receives user input:

- Avoid assumptions about the size, type, or content of the data that users will attempt to send to an application. Consider the following:
 - How will the application behave if an attacker submits a 50-MB ZIP file to a field designed for the entry of a telephone number?
 - How will the application behave if an attacker embeds an ALTER TABLE statement in a text field?

- Ensure that the size and data type of the input are checked.
- Check the content of string variables and ensure that the application accepts only values within expected ranges. The application should automatically discard any entries that contain binary data, escape sequences, and comment characters.
- Ensure that all XML documents entered are validated against an approved schema.
- Ensure that Transact-SQL statements are not directly built from user input.
- Ensure that stored procedures are used to validate all user input.
- If possible, implement multiple layers of validation. For example, have the Web application perform checks on the Web server, and have SQL Server perform validation checks through stored procedures.
- Ensure that the following strings are not accepted in fields from which file names can be constructed: AUX, CLOCK\$, COM1 through COM8, CON, CONFIG\$, LPT1 through LPT8, NUL, and PRN.

MORE INFO SQL injection attacks

For more information about SQL injection attacks, consult the following MSDN articles: msdn2.microsoft.com/en-us/library/ms161953.aspx and msdn.microsoft.com/msdnmag/issues/04/09/SQLInjection/.

Responding to Virus and Worm Attacks

A virus that specifically targets computers running SQL Server 2005 could attempt to infect SQL Server program files, delete database or program files, or replace existing files with files configured to allow an external attacker access to database information. More common than SQL Server-specific viruses are viruses that target the Windows platform. Viruses can log keyboard activity, sending reports back to the virus writer, or they can act as Trojan horses, allowing attackers remote access to a protected network.

Reducing the Likelihood of Viruses and Worms

The only way to be completely sure that a computer will not become infected by viruses is to disable its network connection and glue shut its magnetic and optical media drives. Although these are extreme measures that you probably won't seriously consider, you can still substantially reduce the chance of a virus or worm launching a successful attack against SQL Server 2005. These strategies include the following:

- **Ensure that antivirus software is installed on the Windows operating system.** Ensure that virus definitions remain current and are set to automatically update themselves.

- **Ensure that service packs, updates, and security hotfixes are regularly applied.** Although all service packs, updates, and security hotfixes should be rigorously tested before being applied, the testing procedure should be refined so that an excessive amount of time doesn't pass between the release of an update and its application to the production server. Use the MBSA tool to regularly evaluate which updates have yet to be applied to the production server.
- **Ensure that database mail is properly secured.** This involves ensuring that only specific users, rather than all users, are able to use this functionality. Securing database mail is covered by the practice at the end of this lesson.
- **Ensure that SQL Server is protected by a firewall.** If possible, install a virus scanner that detects and drops problematic traffic as it attempts to traverse the firewall.
- **Ensure that all accounts that interact with SQL Server have strong passwords.** These passwords should be an adequate length and have a combination of numbers, symbols, and variable-case letters.

Responding to an Infection

If a computer running SQL Server 2005 is infected by a virus or worm, you can take certain steps to minimize downtime and recover from the infection. The following guidelines will assist in recovering from an infection:

- **Isolate the server.** Infected servers can infect other servers. Immediately disconnect from the network any server that is found to be infected.
- **Research the infection.** Find out as much about the infection as possible. The more you understand about an infection, the more likely it is that you can defeat it.
- **Stop SQL Server services.** Stopping SQL Server services can help to reduce the impact of the infection. Some viruses rely upon certain services being active. The 2003 Slammer virus relied upon the SQL Server Service Manager in SQL Server 2000 being active.
- **Update virus definitions, and run a complete antivirus scan.** Virus definitions can be updated from removable media. In some situations, it is preferable to boot in a *preinstallation environment*—which is another name for a scaled-down CD-ROM bootable Windows operating system that has a virus scanner installed—and to scan the server that way. Using this method ensures that your virus scanner isn't missing anything because it has become corrupted by the infection.
- **Install service packs, updates, and hotfixes.** Although all updates should be thoroughly tested before being applied to a server, in times of emergency and

infection, organizations might decide that the damage has already been done and things are not going to get worse if the update deployment process is accelerated.

- **Run any tool released by Microsoft for this problem.** Microsoft often releases tools to disinfect servers from particular worm or virus infections. Administrators who have effectively researched the virus are likely to be aware of these tools.
- **Analyze the risk of collateral damage.** Ensure that the infection has not installed anything else nasty on the network. The virus or worm might have propagated to other computers on the network, where it is lying dormant and ready to flare up again after a predefined period.
- **Verify that the server is now uninfected.** Some infections create backup mechanisms to resist being removed. Verify by using the research performed earlier that these elements do not exist on the server. Once you are satisfied, return the server to the production environment.

In some cases, the damage will be so extensive that the only way to cleanse the server is to perform a clean install and restore everything from backup. Although this is a worst-case scenario, it is better than reconnecting the server to the network and having it infect the other SQL Server 2005 computers in the organization.

Quick Check

1. What method can you use to guard against an injection attack when a user must upload an XML document?
2. Provide an example of a strong 8-character password that could be used for an account that interacts with SQL Server 2005.
3. Which tool can you use to evaluate whether a computer running SQL Server 2005 has the most recent hotfixes, updates, and service packs applied?

Quick Check Answer

1. XML documents can be checked by validating them against a schema.
2. A strong password must contain varying case letters, symbols, and numbers.
3. The Microsoft Baseline Security Analyzer (the MBSA tool) can be used to evaluate whether the computer has the most recent hotfixes, updates, and service packs applied.

Responding to Denial of Service Attacks

Denial of service (DoS) attacks work by overloading the server with spurious requests. As the server's resources are strained by fake requests, the server has trouble dealing with the legitimate ones. These requests can be launched at the same time by hundreds of separate computers on the Internet. These computers have generally become infected with a virus or worm and now function as a botnet. A *botnet* is an array of hundreds or even thousands of virus-infected computers that the attacker uses to overload specific hosts on the Internet. The aim of the attack is to deny the service that the target computer provides. If an organization relies on the Internet for its income and customers cannot reach that site because it has suffered a DoS attack, the result can cause financial ruin.

Diagnosing Denial of Service Attacks

If you have configured a server correctly, you can usually detect DoS attacks early and take steps to minimize any possible damage. The symptoms of a DoS attack include the following:

- **A spike in the number of logins** If the server has 500 logins when there are usually only 25, the administrator should suspect that a DoS attack might be in progress.
- **An increase in network traffic** An unusually large amount of network traffic to the server might indicate that a server is experiencing a DoS attack. Performance Monitor alerts can help in bringing unusually high counter readings to the attention of the administrator.
- **A decrease in SQL Server performance** Another example of a DoS attack would be finding a large number of long-running transactions that are issued by the same user but from different connections.
- **An increase in timeouts when applications request connections** If establishing connections because of timeouts becomes difficult and an examination of SQL Server reveals an unusually high number of existing connections, a DoS attack might be in progress.

Using Performance Monitor Alerts

As mentioned earlier in the chapter, you can configure Performance Monitor alerts on the computer running SQL Server 2005 to diagnose denial of service attacks in their early stages. In general, a computer with SQL Server 2005 installed will have a reasonably predictable load placed on it. Administrators will be aware of the maximum values

that performance counters reach during peak server workload, which is why no values are recommended here. You should configure alerts for the following counters:

- **SQLServer:General Statistics\Logins/sec** A significant increase in this value might indicate that a DoS attack is in progress.
- **SQLServer:General Statistics\User Connections** As is the case with logins, a significant increase in the number of User Connections should be treated as a possible precursor to a DoS attack.
- **SQLServer:Transactions\Transactions** A significant increase in the number of transactions might indicate a DoS attack, especially if the transactions are long running, which indicates that SQL Server is not coping with the load placed on it.
- **Network Interface\Bytes Received/sec** A sustained significant increase in the amount of data that the SQL Server computer is receiving might indicate that a DoS attack is in progress.

Responding to a Denial of Service Attack

Because DoS attacks are initiated and sustained beyond the network perimeter, the only way to stop them is to stop the attacker from attacking. DoS attacks might automatically stop if it looks as though the server itself has crashed or become unresponsive. This section discusses options for responding to a DoS attack, including taking the following actions:

- **Pausing the SQL Server service** Restarting the SQL Server service causes all current DoS connections to be disconnected. A paused service might give the impression that the server has crashed, causing an automated DoS attack designed simply to bring down the server to finish. If the DoS attack is targeted at Windows rather than at SQL Server, attempting a reboot of the server might cause a DoS attack to stop for similar reasons.
- **Using a dedicated administrator connection (DAC) to shut down or restart the SQL Server service** Most database administrators manage their servers remotely. Often the servers are in a different location than the administrator, so a DoS attack can make it almost impossible for the administrator to connect to the server to pause or shut down the SQL Server service. If you cannot connect normally, you can connect via DAC to shut down the SQL Server service. The benefit of DAC over technologies such as Emergency Management Services (EMS) is that EMS is more of a blunt instrument. EMS allows you to shut down errant

services, but it might not allow you to do so in a graceful manner. It is possible to connect to DAC using SQLCMD or by using SQL Server Management Studio and typing **ADMIN:** followed by the name of the instance. For example, to connect to DAC on instance MELBOURNE, you would connect to **ADMIN:MELBOURNE** in the Connect To Database Engine dialog box.

- **Configuring a firewall** In some cases, DoS attacks come from a single IP address or a set range of IP addresses. If this is the case, blocking these addresses at the firewall is likely to resolve the problem. Analysis of network traffic usually shows that DoS attack traffic is substantially different from genuine traffic. After having a talented network engineer examine the DoS traffic, a fully featured firewall can be programmed to drop DoS traffic at the network perimeter and allow legitimate traffic to pass through to SQL Server. DoS attacks also tend to target specific public IP addresses rather than DNS names, so shifting a server that is under sustained attack to a new IP address and updating DNS accordingly can resolve the problem.

Responding to Internal Attacks

Internal attacks are more difficult to deal with than external attacks because you cannot block internal attackers by external firewalls. Internal attackers are trusted members of the organization. Their ulterior motives often don't become clear until after they have completed their attack. Internal attackers might be attempting to gain access because they are secretly working for a competitor, they want to do some damage, or they simply are curious and want access to restricted information. The following guidelines discuss responses and precautions that you can take in relation to internal attacks:

- **Disable accounts.** When an employee has been fired or has resigned, immediately disable the employee's access. Ensure that alerts are raised if someone tries to use these credentials for access. Someone who has left the organization has no reason to continue to attempt access other than to attack.
- **Alter application account credentials.** If attacks occur using the organization's application security context, you need to change the account and password for that application.
- **Have users change their passwords regularly.** Ensure that users change their passwords regularly so that an attacker who has learned a user's password can use that knowledge only for a limited amount of time.

- **Increase auditing strength.** The C2 Audit Mode option allows the most complete auditing of the success and failure of access to database objects. The downside of this type of auditing is that it significantly affects the performance of SQL Server 2005. If the audit log directory runs out of space, SQL Server 2005 shuts down. More information about the C2 audit mode can be found at [msdn2.microsoft.com/en-us/library/ms187634\(d=ide\).aspx](http://msdn2.microsoft.com/en-us/library/ms187634(d=ide).aspx).

NOTE Attackers on the DBA team

There is little that you can do when the internal attacker is a fellow member of the database administration team. Anyone with a sufficient level of rights can hide his tracks quite effectively.

PRACTICE Securing Database Mail

Securing database mail is an important part of ensuring that SQL Server 2005 cannot be used to propagate viruses and worms. To configure and secure database mail, perform the following steps:

1. Open SQL Server Management Studio, and connect to the local instance.
2. Expand the Management folder.
3. Right-click Database Mail. Choose Configure Database Mail. This launches the Database Mail Configuration Wizard. Click Next.
4. On the Select Configuration Task page, choose Set Up Database Mail By Performing The Following Tasks and click Next.
5. If you are asked whether you would like to enable Database Mail, click Yes. This opens the New Profile page as shown in Figure 12-5.
6. Enter **Securemail** as the profile name and then click Add to open the New Database Mail Account dialog box.

IMPORTANT Stages in securing database mail

Securing database mail is a two-stage process. It requires that you configure a mail account on the SQL server and that you configure an SMTP server appropriately on the network. This SMTP server should require an SSL connection and SMTP authentication. This authentication should occur using a specially configured account. The organization's firewall should be configured to allow only SMTP traffic from specific and known SMTP servers. That way, a rogue SMTP server can't be used to blast e-mail out over the Internet.

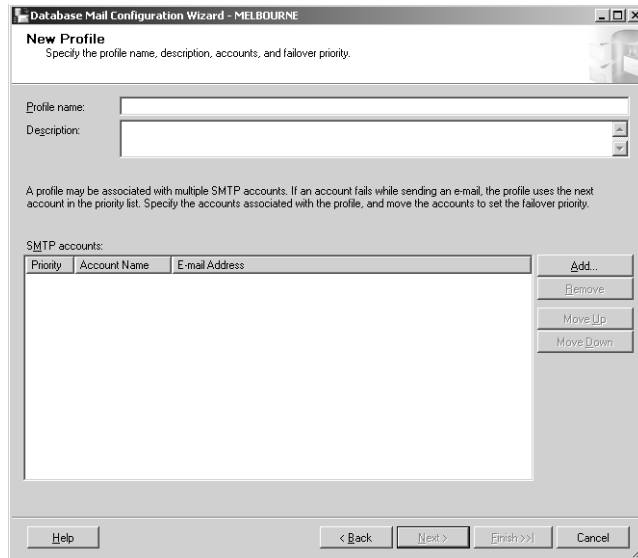


Figure 12-5 Creating a new database mail profile.

7. You should configure the details of the mail account in a way similar to that shown in Figure 12-6. The SMTP server that the database mail account connects to should be configured to accept only SSL connections and should require SMTP authentication using a specially configured account. Click OK to close the New Database Mail Account dialog box. Click Next.

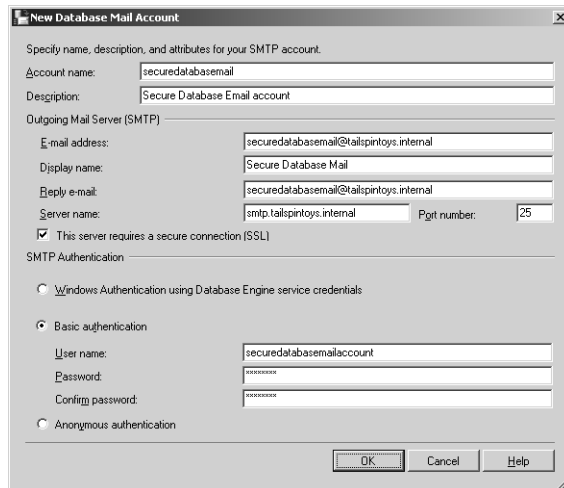


Figure 12-6 Configuring the mail profile to authenticate against an SMTP server.

8. On the Manage Profile Security page, click the Private Profiles tab. Select the user name of a database user to whom you want to grant access to database mail, and then select the Access check box next to Securemail. Click Next.

NOTE The Manage Profile Security page

On this page, you secure database mail by restricting it only to trusted users. When running this practice, only the default NT AUTHORITY\NETWORK SERVICE and NT AUTHORITY\SYSTEM accounts will be available.

9. Accept the defaults on the Configure System Parameters page, and click Next.
10. Click Finish to finish the wizard.

Lesson Summary

- SQL injection attacks involve an attacker executing unauthorized SQL statements against a database by taking advantage of Web applications that fail to properly check user input.
- The best defense against SQL Server injection attacks is validating input before it reaches SQL Server.
- The best defense against virus and worm attacks is ensuring that virus definitions are up to date and that the most recent service packs and hotfixes have been applied.
- Denial of service attacks involve the database server being overloaded with spurious requests. Performance Monitor alerts can be configured to warn of DoS attacks.
- Although it is not possible to stop a DoS attack, blocking attacking IP address ranges and pausing the SQL Server service are two strategies that might halt the attack early.
- Internal attacks are usually carried out by trusted users. Although such attacks are difficult to predict in advance, any users scheduled to resign or be fired should have their user account disabled. Passwords of close colleagues should also be changed in case the user who is leaving is aware of them.

Lesson Review

You can use the following questions to test your knowledge of the information in Lesson 2, “Protecting Against Threats and Attacks.” The questions are also available on the companion CD if you prefer to review them in electronic form.

NOTE Answers

Answers to these questions and explanations of why each answer choice is correct or incorrect are located in the "Answers" section at the end of the book.

1. Your manager informs you that Barry is about to get fired and that the company is concerned he might try to retrieve valuable sales information from the database once he learns of his status. Which of the following steps should you take to ensure this does not happen? (Choose all that apply.)
 - A. Get everyone who has worked with Barry to change their passwords.
 - B. Alter application context account credentials that Barry might be aware of.
 - C. Disable Barry's user account.
 - D. Reboot the computer running SQL Server 2005.
2. The SQL Server 2005 computer on the screened subnet is suffering from a denial of service attack and has become unresponsive to standard remote administration technologies. The technician who manages your organization's perimeter hardware firewall was in a nonfatal car accident this morning, and it will be several hours before the firewall can be reconfigured to deal with the denial-of-service traffic. Your company's servers are located at another facility several miles away. Because you usually ride a bicycle to work, it will take some time for you to gain physical access to the server in question. Which of the following technologies can you use to gracefully shut down the SQL Server service?
 - A. Emergency Management Services (EMS)
 - B. Dedicated administrator connection (DAC)
 - C. Remote procedure call (RPC)
 - D. Remote Desktop Protocol (RDP)
3. You suspect that your SQL Server 2005 server has become infected by a recently released worm. Which of the following steps should you take to deal with this problem? (Choose all that apply.)
 - A. Reboot the computer.
 - B. Isolate the computer from the network.
 - C. Connect to the computer using a dedicated administrator connection (DAC).
 - D. Update antivirus definitions, and rescan the computer.

Chapter Review

To further practice and reinforce the skills you learned in this chapter, you can complete the following tasks:

- Review the chapter summary.
- Review the list of key terms introduced in this chapter.
- Complete the case scenarios. These scenarios set up real-world situations involving the topics of this chapter and ask you to create a solution.
- Complete the suggested practices.
- Take a practice test.

Chapter Summary

- Physical server security is about ensuring that attackers are unable to physically access a server. Server room doors should use two-factor authentication, record a log, and lock when several authentication failures occur in a row.
- Server rooms should always use halon-based fire suppression technology.
- Backup tapes should be stored securely in a fireproof safe.
- SQL Server should never be directly exposed to the Internet. It should always be installed on a network that is protected from the Internet by at least one firewall.
- The best defense against SQL Server injection attacks is validating input before it reaches SQL Server.
- The best defense against virus and worm attacks is ensuring that virus definitions are up to date and that the most recent service packs and hotfixes have been applied.
- Blocking attacking IP address ranges and pausing the SQL Server service are two strategies that can halt denial of service attacks early.

Key Terms

- denial of service (DoS) attack
- halon
- injection attack

- keylogger
- two-factor authentication
- Radio Frequency Identification (RFID)

Case Scenarios

In the following case scenarios, you will apply what you've learned in this chapter. You can find answers to these questions in the “Answers” section at the end of this book.

Case Scenario 1: Physically Securing a Server Room

You are responsible for the security of database servers at Contoso Corporation. Contoso has just moved into a new building. A small room in the basement, which was previously used as a storeroom, will now function as a server room and will host the Contoso database servers. You have been tasked with ensuring that this new server room has adequate physical security. With this in mind, answer the following questions:

1. What will you need to remove from the basement room before installing the servers?
2. What design elements should the door lock include?
3. When the door to the server room is secured, what other possible entry points to the room should you also check?

Case Scenario 2: Responding to a Denial of Service Attack

After your organization was hit by a denial of service attack on its Web server, management decided that the company must develop a response plan for denial of service attacks on all servers critical to the company's operation. As the senior database administrator, you are responsible for developing the denial of service attack response plan for the company's three computers running SQL Server 2005 Enterprise Edition. Developing this denial of service attack response plan involves you knowing the answers to the following questions:

1. Before a server reaches the stage of becoming unresponsive, what tools can you use to diagnose a denial of service attack in its early stages?
2. If the server has become non-responsive to the point where standard tools cannot be used to gracefully shut down SQL Server, what strategy should be pursued?

Suggested Practices

To help you successfully master the exam objectives presented in this chapter, complete the following practice tasks:

Performing a Security Audit of the Existing Security Infrastructure Based on the Security Plan

- **Practice 1: Document physical server security.** Using the points outlined in Lesson 1, “Auditing the Existing Infrastructure,” document improvements that could be made to the physical security of the SQL Server servers in your organization.
- **Practice 2: Create a baseline security analysis.** Use the Security Configuration and Analysis tool to generate a baseline analysis of the security of a SQL Server 2005 computer. If possible, perform the analysis on one of the servers at your organization; otherwise, use one of the practice SQL Server servers.

Prepare for and Respond to Threats and Attacks

- **Practice 1: Disable xp_cmdshell.** Use the appropriate Transact-SQL statements to disable xp_cmdshell on one of your practice SQL Server 2005 computers.
- **Practice 2: Configure a dedicated administrator connection.** Configure and then test a DAC. From a remote computer, connect to the server that you configured with a DAC.

Take a Practice Test

The practice tests on this book’s companion CD offer many options. For example, you can test yourself on just one exam objective, or you can test yourself on all the 70-444 certification exam content. You can set up the test so that it closely simulates the experience of taking a certification exam, or you can set it up in study mode so that you can look at the correct answers and explanations after you answer each question.

MORE INFO Practice tests

For details about all the practice test options available, see “How to Use the Practice Tests” in this book’s Introduction.

Appendix

A 180-day evaluation edition of the Microsoft SQL Server 2005 Enterprise Edition software is included with this book. Using this evaluation software, and other evaluation software that you can download, you can build a test environment on which you can perform all the practical exercises in this book. There are two options for creating a test lab. The first is to use two physical computers and install the operating system and database software on these computers. A cheaper alternative is to download a copy of Microsoft Virtual Server 2005 R2 from the Microsoft Web site and create two separate virtual servers.

NOTE Virtual Server 2005 R2

Virtual Server 2005 R2 can be downloaded from the Microsoft Virtual Server Web site at <http://www.microsoft.com/windowserversystem/virtualserver/default.aspx>.

Each virtual server will consume approximately 10 gigabytes (GB) of disk space and 700 megabytes (MB) of RAM. The minimum specification for the virtual configuration outlined in this appendix is as follows:

- Pentium IV or equivalent
- 1.5 GB of RAM
- 20 GB hard disk drive space

If you decide not to use Virtual Server 2005 R2 and instead use separate stand-alone computers on their own hardware, the minimum specification will be that of SQL Server 2005 Enterprise Edition, which is as follows:

- 600 Mhz or faster Pentium III
- 512 MB of RAM
- Approximately 5 GB hard disk drive space

NOTE RAM expense

The price of RAM is so low these days that adding the capacity to run several extra servers at once using virtual machine software can cost less than a monitor for a second computer!

Configuring the Computers

Configuring the computers requires the following steps:

- Installing and configuring the Windows Server 2003 R2 180-day evaluation software
- Installing the SQL Server 2005 Enterprise Edition 180-day evaluation software
- Installing SQL Server 2005 SP1
- Installing sample databases

Installing and Configuring the Windows Server 2003 R2 180-Day Evaluation Software

NOTE Obtaining the Windows Server 2003 R2 evaluation software

You can obtain the Windows Server 2003 R2 180-day evaluation software from the following location:
<http://www.microsoft.com/windowsserver2003/evaluation/trial/default.aspx>

Configure both computers in the following manner:

1. Install Windows Server 2003 R2 using the default configuration. You should set the computer names to Melbourne and Glasgow. Configure the Administrator account with the password **P@ssw0rd**.

NOTE Installing Windows Server 2003 R2

If you are unsure about how to install Windows Server 2003, read the following Microsoft TechNet article: <http://technet2.microsoft.com/WindowsServer/en/Library/c68efa05-c31e-42c9aed6-0391130ceac21033.aspx>.

2. Prior to setting up the lab network configuration, activate each version of Windows Server 2003 R2. When activation is completed, reconfigure the TCP/IP configuration to the following settings:

	Computer 1	Computer 2
Name	Melbourne	Glasgow
IP Address	10.0.0.1	10.0.0.2
Subnet Mask	255.255.255.0	255.255.255.0
DNS Server Address	10.0.0.1	10.0.0.1

3. Configure server Melbourne as a domain controller by running `dcpromo.exe`. Configure the domain with the following settings:
 - Domain controller for a new domain
 - Domain in a new forest
 - Full DNS Name for the new domain: **tailspintoys.internal**
 - Default Database and Log Folders
 - Default SYSVOL folder
 - DNS Registration Diagnostics: Install and configure the DNS server on this computer, and set this computer to use this DNS server as its preferred DNS server
 - Permissions compatible only with Windows 2000 or Windows Server 2003 operating systems
 - Restore mode password: **P@ssw0rd**

NOTE Installing a domain controller

If you are new to the process of installing a domain controller, read the following article for guidance: <http://www.microsoft.com/technet/prodtechnol/windowsserver2003/technologies/directory/activedirectory/stepbystep/domcntrl.msp>.

4. Join computer Glasgow to the `tailspintoys.internal` domain.

NOTE Joining a computer to the domain

If you are unsure about how to join a computer to a domain, consult the following article: <http://technet2.microsoft.com/WindowsServer/en/library/f2d5a706-91a2-4970-bebf66dc187b59c41033.msp?mfr=true>.

5. Install the following components on both computers using the Add/Remove Windows Components section of the Add Or Remove Programs item in the Control Panel:
 - ASP.NET. This item is located under the Application Server category.
 - .NET Framework 2.

Installing SQL Server 2005 Enterprise Edition 180-Day Evaluation

Install SQL Server 2005 Enterprise Edition on both computers by performing the following steps:

1. Insert the SQL Server 2005 Enterprise Edition evaluation media.

2. Click the text that includes Server Components under Install to start the installation.
3. Accept the End User License Agreement. Click Next.
4. Click Install.
5. Click Next three times.
6. Enter the Product Key and click Next.
7. Select the following components to install:
 - SQL Server Database Services
 - Reporting Services
 - Integration Services
 - Workstation components, Books Online and Development Tools
8. Click Next.
9. Leave the Default instance selected, and click Next.
10. On the Service Account page, select Use The Built-in System Account. Click Next.
11. On the Authentication Mode page, leave Windows Authentication selected and click Next.
12. Continue to click Next, accepting default values, until you reach Install. Click Install.

Depending on the speed of your computer, the process will take between 5 minutes to half an hour to complete.
13. Click Next and then click Finish.

Installing SQL Server 2005 Service Pack 1

After SQL Server 2005 is installed, you need to install Service Pack 1.

NOTE Downloading SQL Server 2005 SP1

SQL Server 2005 Service Pack 1 can be downloaded from the following location: <http://www.microsoft.com/sql/sp1.msp>.

After you have obtained this Service Pack, perform the following steps on each computer:

1. Double-click the file SQLServer2005SP1-KB913090-x86-ENU.exe.
2. On the SQL Server 2005 Service Pack 1 Setup Page, click Next.

3. Accept the licensing terms and conditions. Click Next.
4. Continue to click Next until you reach Install. Click Install.
5. If you receive a message about pending updates, click Yes to continue.
6. Click Continue if you receive a message about write locks.
7. When the service pack installation finishes, reboot the computer.

Installing Sample Databases

The practical exercises in this text use several sample databases that can be downloaded from the following Web page:

<http://www.microsoft.com/downloads/details.aspx?FamilyID=e719ecf7-9f46-4312-af896ad8702e4e6e&DisplayLang=en>

Download these files:

- SqlServerSamples.msi
- AdventureWorksDB.msi
- AdventureWorksBI.msi

The install process is the same for each of the previous files. They should be installed in the order listed here. To install these files, perform the following steps:

1. Double-click the file.
2. Click Next at the introductory wizard page.
3. Accept the terms of the license agreement.
4. Accept the default destination folder.
5. Click Next and then click Install.
6. When the file has finished installing, click Finish.

Glossary

- ad hoc query** A query that is run once to elicit specific information.
- AFTER trigger** A trigger that causes an action to occur after an INSERT, UPDATE, or DELETE statement has acted on a table.
- alias data type** A user-created type that is based on one of the existing Transact-SQL data types.
- article** A database object that is included in a publication. A publication can contain different types of articles, including tables, views, stored procedures, and other objects. When tables are published as articles, filters can be used to restrict the columns and rows of the data sent to Subscribers.
- baseline** A series of measurements taken when a system is newly configured or has been extensively reconfigured. Baselines are taken during peak and quiet times and for typical operation. Subsequent measurements are compared with the relevant baseline.
- BLOB** A binary large object. A BLOB is not usually restricted in content type, and content can be several gigabytes in size. BLOB fields are normally used to store graphics, audio, video, or documents.
- canonical row** A row in a database table that Fuzzy Grouping compares to other rows that might contain corrupt data. The canonical row has a score of 1.
- certificate authority** Any entity (individual, department, company, or organization) that issues digital certificates to verify the identity of users, applications, or organizations.
- CHECK constraint** A constraint that allows only values that meet a certain set of criteria to be entered in a column.
- checkpoints** A container or task within an SSIS package from which package execution can be restarted.
- checksum** A form of redundancy check used to protect the integrity of replicated data. It works by performing a series of operations on the original data and storing the resultant value. When the same operations are carried out on replicated data and the checksum obtained is the same as the original checksum, the data is probably not corrupt.
- clustered index** In a clustered index, the leaf-level (the lowest level) is the data. If a table has a clustered index, the data is stored in the order of the index.

- collation** A set of rules that determines how data is compared, ordered, and presented.
- concurrency** The ability of multiple users to access data at the same time.
- configurations** Used to automatically update the values of package properties and package objects at run time.
- conflict resolution policy** A policy that determines how a Publisher reacts when a conflict has been detected.
- control flow elements** A collection of tasks used to manipulate files or the database.
- credentials** Information required from users who want to log in to a network and access its resources.
- cryptography** The process of securely transmitting data over a network in such a way that if the data is intercepted, it cannot be read by unauthorized users.
- DAC (dedicated administrator connection)** A special diagnostic connection that can be used by administrators when other connections are unavailable.
- data definition language (DDL)** A language used to define all attributes and properties of a database.
- data flow elements** A collection of tasks used to transform data.
- data mining** The process of automatically searching large volumes of data for patterns.
- database mirroring** Immediately reproducing every update to a read-write database (the principal database) onto a read-only mirror of that database (the mirror database) residing on a separate instance of the database engine (the mirror server). In production environments, the mirror server is on another computer.
- DDL (data definition language)** A language used to define all attributes and properties of a database.
- DDL trigger** A programmatic device that executes stored procedures in response to Transact-SQL statements that start with a particular set of keywords.
- deadlock** Occurs when two processes are waiting for a resource and neither process can advance because the other process is preventing it from getting the resource.
- dedicated administrator connection (DAC)** A special diagnostic connection that can be used by administrators when other connections are unavailable.
- DEFAULT definition** A method of ensuring that a particular value is entered into a column if no value is provided.
- delegation** A process by which a client can connect one instance of SQL Server to another by having his or her credentials forwarded.

- denial of service (DoS) attack** An attack in which a server is flooded with spurious requests, reducing the server's availability.
- dependencies** The services and drivers that a service requires to be active before the service runs.
- differential backup** A type of backup where only the extents that have changed since the last full backup are archived.
- differential base** A set of files or filegroups that form a differential backup.
- dirty data** Data that contains inaccuracies because of misspellings, truncations, missing or inserted tokens, null fields, unexpected abbreviations, and other irregularities.
- distance function** A function used by Fuzzy Lookup and Fuzzy Grouping that takes account of the edit distance, the number of tokens, the token order, and relative frequencies.
- distributed query** A single query that accesses multiple data sources.
- Distributor** A database instance that acts as a store for replication-specific data associated with one or more Publishers. Each Publisher is associated with a single database (known as a distribution database) at the Distributor.
- DoS (denial of service) attack** An attack in which a server is flooded with spurious requests, reducing the server's availability.
- double hop** A process by which one computer connects to another computer so that it can connect to a third computer.
- endpoint** A SQL Server object that represents the capability for SQL Server to communicate over the network. Each endpoint supports a specific type of communication.
- event category** An event class grouping.
- event class** A mechanism that specifies sets of events by a single value. In a SQL trace, recorded events are instances of the event classes in the trace definition.
- extent** Eight contiguous pages. Uniform extents are owned by a single table or an index. Mixed extents are shared by up to eight tables or indexes. An extent is the smallest unit of data that can be backed up during a differential backup.
- Failed Redundancy** A volume status indicating that a volume still works but has had an underlying disk failure that has caused it to lose redundancy protection.
- failover clustering** A high-availability solution that involves two or more computers, termed as nodes, functioning in unison to support a virtual server that hosts a SQL Server instance. If one node fails, another node continues running the instance.

- filter** A way of reducing the amount of information displayed in the Log File Viewer to a manageable level.
- FOREIGN KEY constraint** A constraint that limits the values that can be entered in a column to those that already exist in another column.
- fragmentation** Splitting of pages within a database when data is added or modified. Fragmented data is no longer contiguous.
- full recovery model** The recovery model that allows the widest variety of restoration possibilities, including point-in-time and named transaction restore. It involves backing up both the database and transaction log.
- fuzzy grouping** A data-cleaning methodology that examines values in a dataset and identifies groups of related data rows and the one data row that is the canonical representation of the group.
- fuzzy matching** A lookup methodology that uses fuzzy matching to locate similar data values in a reference table.
- halon** A gas that disrupts combustion. Halon leaves no residue, is rated safe for human exposure, and is extremely effective at extinguishing flammable liquid and electrical fires.
- handle** When a query is executed as part of a batch, this is a binary hash of the batch's text.
- heap** An unordered structure that stores data rows when a table has no clustered index.
- hot standby server** A server that maintains an almost perfect copy of a production database and that can take over the production role if the primary server fails.
- index** A database index is a pointer to data in a table. It directs a query to the exact physical location of data in that table.
- indexed view** A view that has a unique clustered index. Indexed views exist in a database as rows that realize the view.
- injection attack** An attack in which an attacker inserts SQL code into an input box in an attempt to execute an unauthorized transaction against the server.
- INSTEAD OF trigger** A trigger that causes an action to occur instead of the original action in the transaction.
- isolation level** Controls the locking and row versioning behavior of Transact-SQL statements issued by a connection to SQL Server.

- keylogger** A device that records all keystrokes made on a keyboard. It is often used to gather user name and password information.
- latency** The amount of time that elapses between the completion of a data change at one server and the appearance of that change at another server (for example, the time between when a change is made at a Publisher and when it appears at the Subscriber).
- lock** Before a transaction reads or modifies data, it must protect itself from the effects of another transaction modifying the same data. It does this by requesting a lock on the piece of data.
- Log File Viewer** A tool that allows you to view logs from SQL Server, SQL Server Agent, and Database Mail, as well as view the Windows NT event logs.
- log sequence number** The number that uniquely identifies every record in the Microsoft SQL Server transaction log.
- logging** Writing data about the execution of a package, task, or container to a file or SQL Server table.
- merge replication** A type of replication that typically starts with a snapshot of the publication database objects and data. Subsequent data changes and schema modifications made at the Publisher and Subscribers are tracked with triggers. The Subscriber synchronizes with the Publisher when connected to the network and exchanges all rows that have changed between the Publisher and Subscriber since the last time synchronization occurred.
- metadata** High-level information about data. It's used to enable a user, application, or service to locate a specific item or list of items that meet particular criteria.
- metric** A system of measurement.
- Microsoft Cluster Server (MSCS) service** Implements failover clustering on Windows servers.
- mining model** An object that contains the definition of a data mining process and the results of the training activity. For example, a data mining model might specify the input, output, algorithm, and other properties of the process and hold the information gathered during the training activity.
- mirror server** A server that has an almost up-to-date copy of a production database. The principal server sends transactions to mirror servers, which then immediately apply these transactions.
- Missing** Status assigned to a disk that has become corrupted or disconnected.

- monitoring** Taking periodic snapshots of current performance to isolate processes that could cause problems, and gathering information continuously over time to track performance trends.
- multibased differential** A differential backup of more than one primary database backup bases.
- node** A server in a cluster.
- nonclustered index** In a nonclustered index, the leaves contain bookmarks to the actual data. A nonclustered index does not specify the order in which data is stored.
- notification** Executes in response to a variety of Transact-SQL DDL statements and SQL Trace events by sending information about these events to a Service Broker service.
- nullability** A characteristic that determines whether a column accepts the value NULL.
- Offline** Status assigned to a disk that is only intermittently available.
- OLE DB** An API used to access data stored in any format for which an OLE DB provider is available.
- online analytical processing (OLAP)** A technology that allows users to perform sophisticated data analysis on typically large amounts of enterprise data to gain insight on the information it contains.
- Open Database Connectivity (ODBC)** A data-access API that supports access to any data source for which an ODBC driver is available.
- Package Deployment utility** A collection of files that can be copied to a new computer running SQL Server 2005 and used to install a completed package.
- page** The primary unit of SQL Server data storage. Pages are 8 KB in length. A page is the smallest unit of data that can be restored.
- paging** The process by which data pages are transferred from random access memory (RAM) to virtual memory on the hard disk.
- partitioning** Splitting a large table into smaller, individual tables so that queries accessing only a subset of the data can run faster because there is less data to scan.
- performance alert** A procedure that can be configured to write an event to an applications log, send an administrative message, and start an executable program. Alerts are triggered by the values in performance counters.

- performance counter log** Periodically scans selected performance counters and records their values over a period of time.
- performance trace log** Records performance data whenever events related to its source provider occurs.
- PRIMARY KEY constraint** A constraint that is similar to a uniqueness constraint except that it can involve more than one column.
- principal server** The server that holds the production database in a mirrored configuration.
- privileges** Rights and permissions to an object that are granted implicitly or explicitly to a security principal such as a user account.
- publication** A collection of one or more articles from one database. The grouping of multiple articles into a publication makes it easier to specify a logically related set of database objects and data that are replicated as a unit.
- Publisher** A database instance that makes data available to other locations through replication. The Publisher can have one or more publications, each defining a logically related set of objects and data to replicate.
- query** A statement or series of statements that runs against a database and returns information retrieved by the database.
- Queue Reader Agent** A SQL Server 2005 subsystem that detects conflicts.
- quiescing** Stopping all activity and verifying that all nodes have received all outstanding changes.
- Radio Frequency Identification (RFID)** A technology that allows for the transmission of identity information using radio waves.
- recovery point** A specific point in time, transaction, or log sequence number.
- replication agent** A stand-alone program that replication uses to carry out the tasks associated with tracking changes and distributing data. By default, replication agents run as jobs scheduled under SQL Server Agent.
- report definition** An .rdl file that contains information about the query and layout of a report.
- report snapshot** A report that contains data captured at a specific point in time.
- resource group** A combination of nodes and shared disks.
- restore sequence** A set of Transact-SQL statements that will bring a database back to a particular recovery point.

- rolling forward** The process of applying logged changes to data in a database to bring it forward to a target recovery point.
- SCSI (small computer system interface)** A general-purpose I/O bus.
- securable** A SQL Server 2005 object that has associated permissions that can be granted to a security principal.
- security context** Credentials supplied by a user account so that an application or service can obtain access to resources.
- security principal** An object, such as a user or group account or a server-level or database-level role, which has a set of permissions that determine its level of access to securables.
- self mapping** A default technique whereby current security credentials are impersonated for use in resolving distributed queries.
- Service Broker** Provides mechanisms for automatically starting programs that process a queue when there is useful work for the program to do, and provides queuing as an integral part of the database engine. Service Broker can be used, for example, to implement notifications that inform you when services have started.
- Severity** A code assigned to an error message that indicates the level of impact a failure has had.
- small computer system interface (SCSI)** A general-purpose I/O bus.
- snapshot replication** A type of replication that distributes data exactly as it appears at a specific moment in time and does not monitor for updates to the data. When synchronization occurs, the entire snapshot is generated and sent to Subscribers.
- source provider** An application or operating system service that has traceable events.
- sparse files** Files that host snapshot data.
- SQL Server log** Detects current or potential problem areas, and displays automatic recovery messages.
- SSIS variables** Allow the package to be dynamically controlled at run time. SSIS variables include items such as connection strings.
- Subscriber** A database instance that receives replicated data. A Subscriber can receive data from multiple Publishers and publications. Depending on the type of replication chosen, the Subscriber can also pass data changes back to the Publisher or republish the data to other Subscribers.

- subscription** A request for a copy of a publication to be delivered to a Subscriber. The subscription defines what publication will be received, as well as where and when it will be received. A subscription can be push or pull.
- threshold value** A predefined value that initiates an action if a counter either exceeds it or falls below it. Typically, a counter reaching a threshold value triggers an alert.
- timestamp** Logged with an event description to specify when the event occurred.
- trace flag** Temporarily sets specific server characteristics or switches off a particular behavior. Trace flags are used to diagnose performance issues or to debug stored procedures.
- tracer token** Provides a method of measuring latency in transactional replication and of validating the connections between the Publisher, Distributor, and Subscribers. SQL Server 2005 replication writes a token (a small amount of data) to the transaction log of the publication database, marks it as though it were a typical replicated transaction, and sends it through the system.
- training** Populating the empty structure of a mining model with the patterns that describe the model.
- transaction** A collection of tasks where changes made can be rolled back if all tasks in the collection do not execute.
- transaction log** A technology that provides fault tolerance and crash recovery for critical database files. Transaction logs record all the operations that are run against the database, both complete (committed) and incomplete. They play an important role in providing fault tolerance and recoverability for databases.
- transactional replication** A type of replication that typically starts with a snapshot of the publication database objects and data. As soon as the initial snapshot is taken, subsequent data changes and schema modifications made at the Publisher are typically delivered to the Subscriber as they occur (in near real time). The data changes are applied to the Subscriber in the same order and within the same transaction boundaries as they occurred at the Publisher.
- Transact-SQL data type** One of the default data types that ship with SQL Server 2005.
- truncation** The removal of nonactive portions of the transaction log. Truncation is almost always achieved through the backup of the transaction log.
- two-factor authentication** An approach to security in which two independent systems are used to verify a user's identity.

UNIQUE constraint A constraint that stops duplicate values from being entered in a column.

user-defined data type A method of extending the scalar type system to allow the storage of CLR objects within the database.

virtual server A resource group, its network name, and its IP address.

witness server Monitors principal and secondary servers involved in database mirroring and initiates the automatic failover process if the principal server becomes unresponsive.

Answers

Chapter 1: Lesson Review Answers

Lesson 1

1. **Correct Answer: D**
 - A. **Incorrect:** Alerts allow you to perform an immediate action or start a program or procedure. Configuring alerts does not enable you to record and analyze server activity over a 24-hour period.
 - B. **Incorrect:** System Monitor in graph mode lets you take a snapshot of server activity. It does not enable you to record and analyze server activity over a 24-hour period.
 - C. **Incorrect:** Task Manager lets you observe memory, network, and processor usage. It does not enable you to record and analyze server activity over a 24-hour period.
 - D. **Correct:** A counter log enables you to record server activity over a 24-hour period. Saving the log as either a comma-delimited text file or a SQL Server database file enables you to analyze the information.
2. **Correct Answer: C**
 - A. **Incorrect:** It is probably a good idea to do this if you have any hard disks that could run low on storage space, but it does not address the problem described.
 - B. **Incorrect:** The cache hit ratio determines whether requests are being satisfied from cache memory. It is used in addition to the Memory object counters to diagnose pressure on the memory resource. Lack of available memory is not the cause of this problem.
 - C. **Correct:** Memory, CPU, disk, and processor resources are not under stress. Client applications that connect to SQLSRVA over the network are likely to be running slowly as a result of network delays. The applications are running slowly throughout the working day, so you do not need to create a counter log to capture unacceptable counter levels. You could also use Task Manager to get a snapshot of network usage, but the question does not offer this option.

- D. **Incorrect:** You do not need to create a counter log, and the counters specified in this answer are not used for detecting network congestion.
3. **Correct Answer: D**
- A. **Incorrect:** Although the pages/sec reading indicates pressure on the RAM resource, the buffer cache hit ratio indicates that this pressure is probably not affecting SQL Server operations. Before adding more RAM, you need to find out what is causing memory pressure.
- B. **Incorrect:** The question gives no indication that the processor resource is under stress. You need to perform further checks before spending money on hardware.
- C. **Incorrect:** The Pages/sec counter value indicates that memory is under pressure. Under these circumstances, you probably would detect a high number of disk I/O operations, but these would be the result of excessive paging arising from memory stress.
- D. **Correct:** You should move any non-SQL Server applications to another server. In general, SQL Server 2005 should be the only major application on a physical server.
4. **Correct Answer: B**
- A. **Incorrect:** A trace log does not alert you when a suspected attack is taking place. Monitoring network counter values tells you the volume of network traffic but gives you no other information.
- B. **Correct:** An alert, triggered by a high level of network traffic, can warn you when a suspected attack is taking place. Configuring the alert to start a Network Monitor capture enables you to audit and analyze network traffic.
- C. **Incorrect:** A counter log does not warn you when a high volume of network traffic occurs. It records the level of network traffic but does not enable you to audit and analyze that traffic.
- D. **Incorrect:** If you check network traffic periodically, you might miss an incidence of high network traffic. Task Manager gives you an instant snapshot but does not record network traffic, and it does not enable you to audit and analyze that traffic. Also, this procedure requires excessive administrative effort.

Lesson 2

1. Correct Answer: B

- A. **Incorrect:** You are interested in disk I/O activity, not CPU activity.
- B. **Correct:** You want to find out what events are reading data from or writing data to the volume a large number of times.
- C. **Incorrect:** You are interested in how often an event accesses the disk, not how long it lasts.
- D. **Incorrect:** The identity of the server process does not have a direct bearing on disk I/O activity.

2. Correct Answer: B

- A. **Incorrect:** The *sys.dm_db_task_space_usage* DMV identifies large queries, temporary tables, or table variables that are using a large amount of tempdb disk space. It does not address locking issues.
- B. **Correct:** The *sys.dm_tran_locks* DMV returns information about currently active lock manager resources. You can use it to obtain further information when you suspect locking is a problem.
- C. **Incorrect:** The internal clock hand controls the size of a cache relative to other caches. The external clock hand starts to move when SQL Server as a whole is experiencing memory pressure. Information about clock hand movements is exposed through the *sys.dm_os_memory_cache_clock_hands* DMV. The DMV does not address locking issues.
- D. **Incorrect:** You run Profiler and examine the SQL: StmtRecompile event class to determine which stored procedures are recompiled. The event does not address locking issues.

3. Correct Answer: C

- A. **Incorrect:** The buffer cache hit ratio counter indicates how often a query obtains information from cache rather than accessing hard disk. An unacceptable value in this counter would indicate that queries could be performing slowly because of memory pressure. However, monitoring this counter will not provide the statistics required for the SLA.
- B. **Incorrect:** The SQL: StmtRecompile event class indicates which stored procedures and statements have been recompiled. Excessive recompilation can put pressure on a server's CPU resource and affect query performance.

However, creating a Profiler trace by using this event class will not provide the statistics required for the SLA.

- C. **Correct:** Using a Profiler template ensures consistency in the way your team collects query response statistics.
- D. **Incorrect:** Your team members can use the *sys.dm_db_session_space_usage* and *sys.dm_db_task_space_usage* DMVs to identify large queries, temporary tables, or table variables that are using a large amount of tempdb disk space. However, these DMVs will not provide the statistics required for the SLA.

Lesson 3

1. Correct Answer: A

- A. **Correct:** The Deadlock graph event group presents a graphical description of the tasks and resources involved in a deadlock.
- B. **Incorrect:** The SQLServerLocks: Number of Deadlocks/sec counter returns the number of requests per second that resulted in a deadlock. It does not tell you the cause of a deadlock.
- C. **Incorrect:** The *sys.dm_tran_locks* DMV provides information about lock manager resources. It does not tell you the cause of a deadlock.
- D. **Incorrect:** Running the DTA and implementing its recommendations should improve query performance. It does not tell you the cause of a deadlock.

2. Correct Answer: B

- A. **Incorrect:** The Log Cache Hit Ratio counter returns the percentage of log cache reads satisfied from the log cache. It does not tell you whether the log is running out of space.
- B. **Correct:** The Percent Log Used counter returns the percentage of space in the log that is in use.
- C. **Incorrect:** The Log Growths counter returns the total number of times the transaction log for the database has been expanded. It does not indicate whether the log is running out of space.
- D. **Incorrect:** The Log Shrinks counter returns the total number of times the transaction log for the database has been shrunk. It does not indicate whether the log is running out of space.

Chapter 1: Case Scenario Answers

Case Scenario: Resolving Physical Server and Database Bottlenecks

1. The memory subsystem is under stress. The average Pages/sec value should not exceed 20, even during periods of high usage. The Buffer Cache Hit Ratio counter indicates that the bottleneck in the memory subsystem is affecting SQL Server operation. The value in this counter should ideally be greater than 99 percent and should not be less than 90 percent. The % Disk Time counter value is less than 55 percent, the Avg. Disk Queue Length is 2 (in a six-disk setup, any value under 12 is acceptable), and % Processor Time is less than 80 percent. The disk and CPU resources are not under pressure.
2. You should include the /3GB switch in the boot.ini file. Memory is currently the bottleneck in the system. By enabling the /3GB switch, you will give SQL Server access to 3 GB of the available 4 GB of memory. You should add more RAM as a long-term solution.
3. The graphical execution plan in SSMS enables you to analyze a query plan and receive assistance to improve the query performance. You can use SQL Server Profiler to capture a trace when a script containing the queries you want to analyze is running, or when you execute the queries directly by typing them into the Query Editor. You can use the DTA to tune databases and create an optimized query environment.
4. The SQL Server: SQL Statistics object provides counters to monitor compilation. If these counters indicate a high number of recompiles, you then need to look at the SP:Recompile and SQL: StmtRecompile event classes in the Profiler trace to determine which stored procedures are recompiled.
5. You can use the Deadlock graph event group in SQL Server Profiler. This presents a graphical description of the tasks and resources involved in a deadlock.
6. You can use the SQLServer:Databases:PercentLogUsed counter to trigger a SQL Server performance condition alert. You could, for example, use this alert to start a transaction log backup.

Chapter 2: Lesson Review Answers

Lesson 1

1. **Correct Answer: A**
 - A. **Correct:** The *sys.dm_exec_query_stats* DMV returns aggregate performance statistics for cached query plans. These statistics include the usage of CPU and disk I/O resources.
 - B. **Incorrect:** The *sys.dm_db_index_physical_stats* DMV returns size and fragmentation information for the data and indexes of the specified table or view. It does not return resource usage statistics.
 - C. **Incorrect:** You can add Showplan event classes to a trace definition so that Profiler gathers and displays query plan information in the trace. This procedure does not return resource usage statistics.
 - D. **Incorrect:** The *sqlcmd* utility allows you to enter commands, including operating system commands, by using the Command Console. It does not return resource usage statistics.
2. **Correct Answer: C**
 - A. **Incorrect:** Restarting the SQL Server computer has an impact on database users. Resetting the statistics in the DMV has a minimal impact on users.
 - B. **Incorrect:** Restarting the SQL Server service has an impact on database users. Resetting the statistics in the DMV has a minimal impact on users.
 - C. **Correct:** The DMV statistics are cumulative. You need to reset them so that you can determine whether the readings relate to current performance problems. Resetting the DMV statistics has a minimal impact on users.
 - D. **Incorrect:** The DMV statistics are cumulative. Inspecting the value in the *waiting_tasks_count* counter will not help you determine whether the DMV readings relate to current performance problems. You need to reset the statistics.

Lesson 2

1. **Correct Answer: B**
 - A. **Incorrect:** Performing a substring operation to determine the first letter of a name is unnecessary and slows down the query.

- B. **Correct:** This query checks that the name starts with M without performing a substring operation. Data is case insensitive, so 'm%' is the same as 'M%'.
 - C. **Incorrect:** Performing a substring operation to determine the first letter of a name is unnecessary and slows down the query. Duplicate rows probably do not exist and do not, in any case, present a problem, so the DISTINCT operator is unnecessary and slows query operation.
 - D. **Incorrect:** Duplicate rows probably do not exist and do not, in any case, present a problem, so the DISTINCT operator is unnecessary and slows query operation.
2. **Correct Answer: D**
- A. **Incorrect:** Both Profiler and the DTA (especially the DTA) use server resources. Also, this procedure results in a series of recommendations. It does not quickly give you the information you need.
 - B. **Incorrect:** This procedure lets you obtain recommendations for redesigning a query plan without executing the plan. However, this is not what you want to do. The DTA uses server resources, and you should not run it on a production server. Also, this procedure results in a series of recommendations. It does not quickly give you the information you need.
 - C. **Incorrect:** This procedure gives you a graphical representation of the query plan. You can use the procedure to analyze a query plan without running the query, although it can also display information for a query you execute. It does not quickly give you the information you need.
 - D. **Correct:** The DMV quickly gives you the information you need and does not add significantly to the load on the server. The disadvantage of using DMVs is that you need to execute the query, but that is not a problem in this case.

Lesson 3

1. **Correct Answer: B**
- A. **Incorrect:** The ALTER INDEX statement with REORGANIZE replaces DBCC INDEXDEFRAG. In any case, you need to check the index for fragmentation, not reorganize it.
 - B. **Correct:** You can use the system function `sys.dm_db_index_physical_stats` to detect fragmentation in a specific index, all indexes on a table or in an indexed view, all indexes in a database, or all indexes in all databases.

- C. **Incorrect:** You use the sys.indexes catalog view to determine the fill factor value of an index, not the degree of fragmentation.
 - D. **Incorrect:** Using the CREATE INDEX Transact-SQL statement with the DROP_EXISTING clause rebuilds the index. You need to check it for fragmentation.
2. **Correct Answers: B and C**
- A. **Incorrect:** The queries execute most efficiently when the table rows are ordered by the Name column. The Name column index should therefore be clustered.
 - B. **Correct:** The queries execute most efficiently when the table rows are ordered by the Name column. The Name column index should therefore be clustered.
 - C. **Correct:** You should create a composite index on the columns most often accessed by queries that do not have a clustered index.
 - D. **Incorrect:** A table can have only one clustered index.
3. **Correct Answer: C**
- A. **Incorrect:** A standard view is not indexed. It is consequently not persistent and SQL Server must build its result set each time a query runs. The overhead of re-creating the view dynamically adversely affects query performance.
 - B. **Incorrect:** You specify the pad index option when an index is frequently fragmented and suffers a lot of page splits. You cannot specify this option unless you also specify a fill factor other than 0. In any case, the pad index option is not relevant to this scenario.
 - C. **Correct:** An indexed view is persistent. Running queries against an indexed view ensures that only the required columns are used and the overhead of querying two tables is avoided.
 - D. **Incorrect:** You specify a fill factor other than 0 when an index is frequently fragmented and suffers a lot of page splits. This answer is not relevant to this scenario.

Lesson 4

1. **Correct Answer: B**
- A. **Incorrect:** The Errors And Warnings event category includes event classes that are produced when a SQL Server error or warning is returned. It does not provide information about DML operators.

- B. **Correct:** The Performance event category includes event classes that are produced when DML operators (for example, DELETE, INSERT, and UPDATE) execute.
- C. **Incorrect:** Show Plan Statistics is an event class, not an event category. It provides information about query and stored procedure operation.
- D. **Incorrect:** Trace File Properties is a dialog box in Profiler.

Lesson 5

1. Correct Answer: D

- A. **Incorrect:** The *sys.dm_tran_locks* DMV returns information about currently active lock manager resources. When a lock contention has been identified, you can use the *sp_who* stored procedure or Activity Monitor to obtain information about the current user connections, the locks they hold, and the processes that are involved. This procedure probably gives you the information you need, but it involves additional administrative effort.
- B. **Incorrect:** The Number of Deadlocks/sec counter tells you how many locks per second become deadlocks. It does not indicate the causes of the deadlocks.
- C. **Incorrect:** Even if you could identify the application code that was causing the problem, it might not be readily accessible for copying into Query Editor. The DTA gives recommendations to improve query performance. It does not directly identify the causes of deadlocks.
- D. **Correct:** The Deadlock Graph event class provides the easiest method of determining why deadlocks occur.

2. Correct Answer: B

- A. **Incorrect:** The Lock Waits/sec performance counter returns the number of lock requests per second that required the caller to wait. A high value in this counter can indicate locking issues, but the counter is not a good indicator of overall database performance.
- B. **Correct:** The Access Methods object in Microsoft SQL Server provides counters to monitor how the logical data within the database is accessed. The Full Scans/sec counter indicates the number of unrestricted full scans per second. Measured against a baseline, this counter gives an indication of how well databases in a server are performing.

- C. **Incorrect:** The Number of Deadlocks/sec performance counter returns the number of lock requests per second that resulted in a deadlock. A high value in this counter can indicate deadlock problems, but the counter is not a good indicator of overall database performance.
 - D. **Incorrect:** The Transactions/sec counter records transactions that change data, or explicit transactions. The counter is useful for determining whether the number of transactions run against a database has increased substantially, but is not a good indicator of overall database performance.
3. **Correct Answer: C**
- A. **Incorrect:** Read Committed is the default isolation level. It does not permit a transaction to access data that has been modified but not committed by other transactions.
 - B. **Incorrect:** Creating an indexed view is probably a good idea, but it does not permit a transaction to access data that has been modified but not committed by other transactions.
 - C. **Correct:** The Read Uncommitted isolation level allows transactions to read uncommitted modifications (dirty reads), which is what is required in this scenario.
 - D. **Incorrect:** The Repeatable Read isolation level specifies that statements cannot read data that has been modified but not yet committed by other transactions and that no other transactions can modify data that has been read by the current transaction until the current transaction completes.

Chapter 2: Case Scenario Answers

Case Scenario: Dealing with Compatibility Problems and Fragmented Indexes

1. Specify the Read Uncommitted isolation level. Read Uncommitted transactions are not blocked by exclusive locks that would prevent the current transaction from reading rows that have been modified but not committed by other transactions. Because these locks are not required, less memory is needed to manage locks.
2. When you rebuild the index, specify a non-zero fill factor setting—say 75 percent—and set the pad index option to ON. A non-zero fill factor (zero means 100 percent)

leaves space on leaf-level pages so that less splits occur when data is written. Setting the pad index to ON applies the fill factor to intermediate as well as leaf-level pages.

3. Create a plan guide for the query. Plan guides apply query hints to queries in deployed applications when you cannot or do not want to change the application directly. In this case, the plan guide makes the query ignore the unsuitable query hint. If necessary, you can use the plan guide to apply a different query hint.

Chapter 3: Lesson Review Answers

Lesson 1

1. **Correct Answer: B**
 - A. **Incorrect:** If the tempdb database had run out of disk space, other databases on the server also would be experiencing errors.
 - B. **Correct:** If a significant amount of disk space is still left on a volume and the database is behaving as though it has run out of space, it is likely that a limit has been placed on the database file's expansion. You can remove this limit by enabling auto-grow.
 - C. **Incorrect:** In this case, it appears that auto-grow has already been disabled and that the resolution to the problem is to enable it.
 - D. **Incorrect:** In this example, it appears that the database's MAXSIZE has been reached. Reducing this parameter will not fix the problem.
2. **Correct Answer: B**
 - A. **Incorrect:** If the tempdb database was full, different error codes would be issued.
 - B. **Correct:** The SQL Server database engine issues a 9002 error when the transaction log is full.
 - C. **Incorrect:** If the volume hosting the database files was full rather than the transaction log, errors 1101 or 1105 would be visible in the log rather than error 9002. In both cases, however, the database would be unable to process updates.
 - D. **Incorrect:** If the version store was full, error 3959 would be generated rather than error 9002.

Lesson 2

1. Correct Answer: A

- A. **Correct:** A warning icon on a disk in Disk Management indicates that errors have been found on the disk and that the disk has the Online (errors) status.
- B. **Incorrect:** A missing disk does not display a warning icon in Disk Management.
- C. **Incorrect:** An offline disk does not have a warning icon; it has an error icon instead.
- D. **Incorrect:** A foreign disk does not display a warning icon in Disk Management.

2. Correct Answers: A and B

- A. **Correct:** A sudden decrease in performance with all volumes still available suggests a failed RAID-5 redundancy.
- B. **Correct:** One of the disks going offline in a RAID set would cause a failed RAID-5 redundancy.
- C. **Incorrect:** A volume shifting status to Healthy (At Risk) would not cause significant performance degradation.
- D. **Incorrect:** If one of the volumes had failed, it would no longer be present in My Computer.

Lesson 3

1. Correct Answer: A

- A. **Correct:** The Services console can be used to view a service's dependencies.
- B. **Incorrect:** The SQL Server Management Studio cannot be used to view a service's dependencies.
- C. **Incorrect:** The SQL Server Configuration Manager cannot be used to view a service's dependencies.
- D. **Incorrect:** The SQL Server Log File Viewer cannot be used to view a service's dependencies.

2. **Correct Answer: C**

- A. **Incorrect:** The Services console cannot be used to configure error logging for a service.
- B. **Incorrect:** The SQL Server Management Studio cannot be used to configure logging for a service.
- C. **Correct:** Editing the properties of a service through the SQL Server Configuration Manager allows you to configure error logging for a service.
- D. **Incorrect:** The Log File Viewer can be used only to check logs. It cannot be used to configure error logging for a service.

Chapter 3: Case Scenario Answers

Case Scenario 1: Diagnosing Database Configuration Errors

1. If the tempdb database is out of space, it will not be possible to perform queries on the database. The following error codes will be present in the logs: 1101, 1105, 3959, 3967, 3958, and 3966.
2. If the transaction log was full, it would be possible to query the database, but not to update or insert information into the database. Error codes 1101 and 1105 will not be present in the log.
3. If the Mineralogical database has run out of space, error codes 1101 and 1105 will be present in the log and it will not be possible to update or insert information into the database.

Case Scenario 2: Diagnosing Database Hardware Errors

1. Disk Management would reveal a volume with a Failed Redundancy status. One of the disks would have an error and would have a status of Missing. Disk queue length will increase significantly. Performance would decrease significantly.
2. The likely cause of backup jobs and SSIS packages not executing is that the SQL Server Agent service is not running.
3. If there are I/O errors on the disk, the disk will have the status Online (errors). If a volume has experienced I/O errors, it will have the status Healthy (At Risk).

Chapter 4: Lesson Review Answers

Lesson 1

1. **Correct Answer: D**
 - A. **Incorrect:** The server requires at least seven hard disk drives.
 - B. **Incorrect:** The server requires at least seven hard disk drives.
 - C. **Incorrect:** The server requires at least seven hard disk drives.
 - D. **Correct:** Seven disks are required to support this configuration. Two disks, configured using RAID 1, are required for the operating system and SQL Server 2005 program files. Two disks, configured using RAID 1, are required for the transaction log. Three disks, configured using RAID 5, are required for the database files.

2. **Correct Answer: A**
 - A. **Correct:** RAID 0, or disk striping, is not fault-tolerant, and all data on the volume is lost if a hard disk drive that is part of the RAID 0 array fails.
 - B. **Incorrect:** RAID 1, or disk mirroring, is fault tolerant. A RAID 1 volume can survive the loss of a hard disk drive.
 - C. **Incorrect:** RAID 5, or disk striping with parity, is fault-tolerant. A RAID 5 volume can survive the loss of a hard disk drive.
 - D. **Incorrect:** RAID 10 arrays, which are supported by hardware on enterprise database installations, are fault-tolerant and can survive the loss of a hard disk drive.

3. **Correct Answer: D**
 - A. **Incorrect:** If the processor on the first node stops working, the cluster fails over to the second node.
 - B. **Incorrect:** If the disk drive that hosts the operating system on the second node stops working, the cluster fails over to the first node.
 - C. **Incorrect:** If the network interface cards on the first node stop working, the cluster fails over to the second node.
 - D. **Correct:** The weakness of a failover cluster is the shared disks. If the shared disks fail, the instance installed on the cluster's virtual server fails.

4. Correct Answers: A and B

- A. **Correct:** You can implement database mirroring only if you configure the database to use the full recovery model.
- B. **Correct:** For automatic failover to work with database mirroring, you must implement a monitor server.
- C. **Incorrect:** A failover cluster is not necessary to achieve automatic failover with database mirroring.
- D. **Incorrect:** You need to use the high-safety rather than high-performance operating mode to support automatic failover.

Lesson 2

1. Correct Answer: D

- A. **Incorrect:** If program files are corrupt and you are unaware of the original installation options, you should uninstall SQL Server 2005 and then perform a complete reinstall. You will be unable to reach single-user mode if the program files are corrupt.
- B. **Incorrect:** If program files are corrupt and you are unaware of the original installation options, you should uninstall SQL Server 2005 and then perform a complete reinstall. Attempting to rebuild the system databases will not restore corrupt program files.
- C. **Incorrect:** You should not attempt to rebuild and repair corrupt program files unless you are sure of exactly which options SQL Server 2005 was installed using.
- D. **Correct:** If you are unaware of which package options were selected, the recommended course of action is to completely uninstall SQL Server 2005 and then perform a reinstall.

2. Correct Answer: C

- A. **Incorrect:** The simple recovery model will allow recovery only to the most recent backup.
- B. **Incorrect:** The bulk-logged recovery model will not always allow recovery to a named transaction. If a bulk operation has been logged, restoration can occur only to the end of the transaction log.
- C. **Correct:** The full recovery model allows restoration to a named transaction.

3. Correct Answer: B

- A. **Incorrect:** If one file in a filegroup is offline, the entire filegroup is offline. If the primary filegroup is offline, the database is offline.
- B. **Correct:** If one file in a filegroup is offline, the entire filegroup is offline. If the primary filegroup is offline, the database is offline.
- C. **Incorrect:** The primary filegroup cannot be marked read-only. Only secondary filegroups can be marked as read-only.
- D. **Incorrect:** The primary filegroup cannot be compressed. Only secondary filegroups can be compressed.

4. Correct Answer: A

- A. **Correct:** Restoring a database that uses the full recovery model without performing a tail-log backup will almost always result in an error. A database administrator should always attempt this sort of backup prior to trying a database restore.
- B. **Incorrect:** A database administrator should always try to perform a tail-log backup, rather than a bulk log backup, prior to attempting a restore.
- C. **Incorrect:** A database administrator should always try to perform a tail-log backup, rather than a full database backup, prior to attempting a restore.
- D. **Incorrect:** A database administrator should always try to perform a tail-log backup, rather than a differential backup, prior to attempting a restore.

Lesson 3

1. Correct Answer: C

- A. **Incorrect:** It is not necessary prior to performing a restoration to run DBCC CHECKDB. Running this stored procedure after a database has been restored is often a prudent move.
- B. **Incorrect:** Truncating the transaction log will not ensure that the database can be restored as fully as possible.
- C. **Correct:** Because the transaction log is located on a volume that is still intact, the database administrator should perform a tail-log backup so that she can restore the database as completely as possible.
- D. **Incorrect:** It is not possible to make a full database backup of the Mineralogical database because the volume that hosted this database has failed.

2. **Correct Answer: A**
 - A. **Correct:** You must drop all snapshots except the one that you want in order to restore the database to prior to performing that operation.
 - B. **Incorrect:** You do not need to have SQL Server 2005 in single-user mode to restore to a snapshot of the database.
 - C. **Incorrect:** A database can be restored to a snapshot only if it is online.
 - D. **Incorrect:** Running DBCC CHECKDB will not resolve this problem.
3. **Correct Answer: C**
 - A. **Incorrect:** The Mineralogical database wasn't corrupted, so restoring it will not achieve your goals.
 - B. **Incorrect:** You can use the *sp_change_users_login* stored procedure only for users who have accounts authenticated by SQL Server.
 - C. **Correct:** It is necessary for you to create database logins for these users and map them to their Windows accounts using the CREATE LOGIN Transact-SQL statement.
 - D. **Incorrect:** Executing the DBCC CHECKDB stored procedure will not restore the logins.

Lesson 4

1. **Correct Answers: B and C**
 - A. **Incorrect:** Using the full recovery model will not cause a database backup to fail.
 - B. **Correct:** Tail-log backups will be unsuccessful if the database is offline and the log files are damaged.
 - C. **Correct:** Tail-log backups will be unsuccessful if the database is offline and the transaction log contains bulk-logged changes.
 - D. **Incorrect:** Using a RAID 0 volume to host the transaction logs, though not recommended, would not cause a tail-log backup to fail unless one of the disks comprising the volume failed.
2. **Correct Answer: B**
 - A. **Incorrect:** If you use the STOP_ON_ERROR option, the restore option fails when the damaged part of the media is encountered.

- B. **Correct:** You use the `CONTINUE_AFTER_ERROR` option to extract the maximum amount of data from damaged media.
 - C. **Incorrect:** If you use the `CHECKSUM` argument, backup checksums must be verified; otherwise, the restore operation fails.
 - D. **Incorrect:** You use `ERROR_BROKER_CONVERSATIONS` to end all conversations with an error stating either that the database is attached or is restored.
3. **Correct Answers: A, B, and C**
- A. **Correct:** The `DBCC CHECKALLOC` procedure is performed when the `DBCC CHECKDB` procedure is run.
 - B. **Correct:** The `DBCC CHECKTABLE` procedure is performed when the `DBCC CHECKDB` procedure is run.
 - C. **Correct:** The `DBCC CHECKCATALOG` procedure is performed when the `DBCC CHECKDB` procedure is run.
 - D. **Incorrect:** The `DBCC SHOWCONTIG` procedure provides information about fragmentation and is not included as a part of the `DBCC CHECKDB` procedure.
4. **Correct Answer: B**
- A. **Incorrect:** To use the repair options of the `DBCC CHECKDB` command, you need to be running SQL Server 2005 in single-user mode. The `REPAIR_ALLOW_DATA_LOSS` option will not repair the database as thoroughly as the `REPAIR_REBUILD` option.
 - B. **Correct:** To use the repair options of the `DBCC CHECKDB` command, you need to be running SQL Server 2005 in single-user mode. Although it is the slowest option, the `REPAIR_REBUILD` option is the best option for performing data repair on a database.
 - C. **Incorrect:** To use the repair options of the `DBCC CHECKDB` command, you need to be running SQL Server 2005 in single-user mode.
 - D. **Incorrect:** To use the repair options of the `DBCC CHECKDB` command, you need to be running SQL Server 2005 in single-user mode.

Chapter 4: Case Scenario Answers

Case Scenario 1: Ensuring Fault Tolerance

1. You can implement a solution that has four RAID 5 volumes, each set using three drives. Three drives is the minimum required for RAID 5. This arrangement will provide the best performance and redundancy without violating the institutional distaste for hardware RAID-based solutions.
2. The only recovery model that always ensures that a database can be brought back up to a point-in-time restore is the full recovery model. The bulk-logged model can achieve this only if no bulk transactions have occurred since the last full database backup.
3. You could implement database mirroring in this situation.

Case Scenario 2: Backup and Recovery

1. Database mirroring sends active transactions from one server to another; log shipping sends backup logs from one server to another. Database mirroring can be configured for automatic failover, but transaction log shipping cannot.
2. The bulk-logged model would be the most appropriate because it minimally logs bulk transactions.
3. You can inform her that she can use the `CONTINUE_AFTER_ERROR` restore option to extract backup data from tapes that have kinks in them.

Chapter 5: Lesson Review Answers

Lesson 1

1. **Correct Answer: B**
 - A. **Incorrect:** Profiler would create a trace that contained service stop, start, and pause events. It would not, however, automatically send you an e-mail when such an event occurs.
 - B. **Correct:** You can configure an event notification on the Audit Server Starts And Stops event class to send you an e-mail if a service stops, starts, or pauses.

- C. **Incorrect:** The Audit Server Principal Impersonation event class can indicate whether the EXECUTE AS option is used with stored procedures and other commands, but not service stop, start, or pause events. Also, Profiler does not automatically send you an e-mail when a selected event occurs.
 - D. **Incorrect:** The Audit Server Principal Impersonation event class can indicate whether the EXECUTE AS option is used with stored procedures and other commands, but not service stop, start, or pause events.
2. **Correct Answer: A**
- A. **Correct:** This alerts you when the transaction log is using more than 75 percent of the disk space that has been allocated to it. You need to either truncate the log or grow the log file manually.
 - B. **Incorrect:** This alerts you when the transaction log is using only 25 percent or less of the space allocated to it. You do not need to take any action in this case.
 - C. **Incorrect:** This alerts you when only 25 percent or less of the volume that holds all the transaction logs is being used. You do not need to take any action in this case.
 - D. **Incorrect:** This alerts you when the volume that holds all the transaction logs is running out of space. It is probably a good idea to configure this alert, but the scenario requires that you receive an alert when any transaction log is running out of space, not the volume that holds all the transaction logs.
3. **Correct Answers: C and D**
- A. **Incorrect:** The SQLServer:Access Methods: Full Scans/sec counter indicates how many full table or index scans are occurring per second. If this number is significantly higher than the baseline value, application performance might be slow. However, this counter might not capture every application, stored procedure, or ad hoc query that is stressing the server. Capturing all the logins following this alert tells you who is logged in, but not which user is causing the problem.
 - B. **Incorrect:** An event notification on the Audit Database Object Management event class notifies you when a CREATE, ALTER, or DROP statement is executed on a database object. This does not identify the transactions that are stressing the server, nor does it tell you which users are running these transactions.

- C. **Correct:** A SQL trace captures information about resource usage and query duration. It also captures login identities. You can use SQL stored procedures to initiate and configure such a trace.
- D. **Correct:** A SQL trace captures information about resource usage and query duration. It also captures login identities. You can use Profiler to generate such a trace.

4. **Correct Answer: D**

- A. **Incorrect:** The server has only one CPU and, therefore, only one effective instance of Processor: % Privileged Time (the 0 and _total instances are the same). This counter returns the percentage of time the processor spends on execution of Microsoft Windows kernel commands. It cannot identify a service or application that is stressing the CPU.
- B. **Incorrect:** This answer is incorrect for the reasons given in answer A.
- C. **Incorrect:** By default, the Process: % Processor Time returns the total CPU usage for all processes. You have already included this counter in the performance log, and it is currently showing an average value of almost 80 percent.
- D. **Correct:** Services and applications are processes. Monitoring all process instances helps you determine which one is stressing the CPU resources. You can also use Windows Task Manager to obtain a snapshot of which processes are consuming a lot of processor resources, but the question does not list this option as an answer.

5. **Correct Answer: C**

- A. **Incorrect:** Alerts allow you to perform an immediate action or start a program or procedure. Configuring alerts does not enable you to record and analyze server activity over the extended period required by a monitoring strategy.
- B. **Incorrect:** System Monitor in graph mode lets you take a snapshot of server activity. It does not enable you to record and analyze server activity over the extended period required by a monitoring strategy.
- C. **Correct:** A counter log enables you to record server activity over the extended time period required by performance monitoring. Saving the log as a SQL database file lets you analyze the information.

- D. **Incorrect:** Task Manager lets you observe memory, network, and processor usage. It does not enable you to capture server activity.
6. **Correct Answer: B**
- A. **Incorrect:** Trace flag 1222 returns the types of locks that are participating in a deadlock and also the current command affected. However, Microsoft recommends using trace flag 1204 for this purpose. Also, trace flag 1222 is global in scope. Setting it locally has no effect.
- B. **Correct:** Trace flag 1204 returns the types of locks that are participating in a deadlock and also the current command affected. It must be enabled globally. Setting it locally has no effect.
- C. **Incorrect:** The SQLServer:Locks: Number of Deadlocks/sec counter shows the number of locks per second on the SQL Server that become deadlocks. It does not, however, show the types of locks that are participating in a deadlock and the current command affected.
- D. **Incorrect:** The SQLServer:Locks: Lock Waits/sec counter shows the number of locks per second that could not be satisfied immediately and had to wait for resources. It gives no information about deadlocks.

Lesson 2

1. **Correct Answer: C**
- A. **Incorrect:** You can view the SQL Server error log to ensure that backup and restore operations, batch commands, scripts, and processes have completed successfully. However, it does not give you the information you require in this scenario.
- B. **Incorrect:** SQL Server and SQL Server Agent log events in the Windows application log. However, they do not log the information you require in this scenario.
- C. **Correct:** You can use the *sys.dm_exec_query_stats* DMV to find out the last time a particular query was executed, the number of times the query has been executed, and the resources consumed by all invocations of the query plan, as well as the least and greatest amount of CPU consumed by a single invocation of the query plan.
- D. **Incorrect:** The *sys.dm_os_wait_stats* DMV returns information about waits encountered by threads that are in execution. It does not give you the information you require in this scenario.

2. **Correct Answer: A**

- A. **Correct:** The Database event category includes event classes that are produced when data or log files grow or shrink automatically.
- B. **Incorrect:** The Objects event category includes event classes that are produced when database objects are created, opened, closed, dropped, or deleted.
- C. **Incorrect:** The Full Text event category includes event classes that are produced when full-text searches are started, interrupted, or stopped.
- D. **Incorrect:** The Locks event category includes event classes that are produced when a lock is acquired, canceled, released, or has some other action performed on it.

3. **Correct Answer: D**

- A. **Incorrect:** You can inspect the SQL Server error log to ensure that backup and restore operations, batch commands, scripts, and processes have completed successfully. The log does not return CPU usage statistics.
- B. **Incorrect:** The SQL Server Agent error log contains warning messages that provide information about potential problems and error messages that typically require intervention by an administrator. The log does not return CPU usage statistics.
- C. **Incorrect:** The *sys.dm_os_wait_stats* DMV returns information about waits encountered by threads that are in execution. The DMV does not return CPU usage statistics.
- D. **Correct:** You can use Profiler to create a SQL trace log and filter the results to display only events with high CPU usage. You can then inspect the SPID and TransactionID data columns.

4. **Correct Answers: B, C, and E**

- A. **Incorrect:** The Windows security event log records security events—for example, successful or failed logins—for which auditing is configured. It does not contain information specific to the MSCS service.
- B. **Correct:** The cluster log contains verbose information related to the MSCS service.
- C. **Correct:** The MSCS service writes application-related information to the Windows application event log.

- D. **Incorrect:** The SQL Server error log contains information about backup and restore operations, batch commands, scripts, and processes. It does not contain information specific to the MSCS service.
- E. **Correct:** The MSCS service logs errors and events in the Windows system event log.
- F. **Incorrect:** The SQL Server Agent service writes warning and error messages to the SQL Server Agent error log. The log does not contain information specific to the MSCS service.

Chapter 5: Case Scenario Answers

Case Scenario 1: Automating, Monitoring, and Configuring Alerts

1. Configure an alert triggered by a value of 70 percent or greater on all instances of the SQLServer:Databases: Percent Log Used counter.
2. Create a performance alert on the LogicalDisk: % Free Space counter on the logical volume that stores the transaction logs. Have the alert notify your team members when the value in this counter is less than 30 percent. Configure the alert to start a backup job that does a full backup of the transaction logs.
3. Because graphical displays are required, you should configure the log to store data in .cdf format. You can use performance logs stored in a database to generate graphs, but you typically do so by running an application that extracts the data to a .cdf file. In this scenario, storing the logs in a .cdf file directly saves administrative effort.

Case Scenario 2: Identifying Slow and Resource-Intensive Transactions

1. Create a SQL trace using system stored procedures or Profiler. Display traces that have high values in the Reads, Writes, CPU, or Duration data columns. Record the login identity in the LoginName data column.
2. You can configure an event notification on the Audit Server Starts And Stops event class to inform your team members when the SQL Server or SQL Server Agent service stops, starts, or pauses. An event notification on the Audit Database Principal Impersonation event class can be used to inform your team members when the EXECUTE AS statement is used.

3. The SQL Server log records the batch commands and scripts that complete successfully.

Chapter 6: Lesson Review Answers

Lesson 1

1. **Correct Answer: B**
 - A. **Incorrect:** The *sys.triggers* catalog view displays information about triggers that apply only to the current database.
 - B. **Correct:** The *sys.server_triggers* catalog view displays information about DDL triggers that apply to all databases on the instance.
 - C. **Incorrect:** The *sys.database_permissions* catalog view is designed to display security, rather than trigger, information.
 - D. **Incorrect:** The *sys.database_principals* catalog view is designed to display security, rather than trigger, information.
2. **Correct Answers: A and B**
 - A. **Correct:** A trigger that uses DDL_TABLE_EVENTS will fire when any table-related statements are executed.
 - B. **Correct:** A trigger that uses DDL_TABLE_EVENTS will fire when any table-related statements are executed.
 - C. **Incorrect:** A trigger that uses DDL_TABLE_EVENTS will fire only when table-related statements are executed, not when database-related statements are executed.
 - D. **Incorrect:** A trigger that uses DDL_TABLE_EVENTS will fire only when table-related statements are executed, not when database-related statements are executed.
3. **Correct Answer: A**
 - A. **Correct:** The first thing a database administrator should try if problems occur after the installation of a service pack is to roll back the service pack.
 - B. **Incorrect:** Performing Automated System Recovery should be done as a last resort. It should not be the first thing database administrators attempt when they encounter problems.

- C. **Incorrect:** Restoring system databases is unlikely to resolve a problem caused by the installation of a service pack.
 - D. **Incorrect:** Restoring user databases is unlikely to resolve a problem caused by the installation of a service pack.
4. **Correct Answer: C**
- A. **Incorrect:** SSMS cannot be used to determine whether a Windows Server 2003 computer has the latest hotfixes, security updates, and service packs applied.
 - B. **Incorrect:** WSUS cannot be used to determine whether a Windows Server 2003 computer has the latest hotfixes, security updates, and service packs applied.
 - C. **Correct:** The MBSA tool can be used to check whether a Windows Server 2003 computer has the latest hotfixes, security updates, and service packs applied.
 - D. **Incorrect:** BIDS cannot be used to determine whether a Windows Server 2003 computer has the latest hotfixes, security updates, and service packs applied.

Lesson 2

1. **Correct Answer: B**
- A. **Incorrect:** The Shrink Database task frees up space in the database; it cannot be used to compact a clustered index.
 - B. **Correct:** The Reorganize Index task can be used to compact a clustered index.
 - C. **Incorrect:** The Check Database Integrity task checks for errors in the database; it cannot be used to compact a clustered index.
 - D. **Incorrect:** The Update Statistics task ensures that the query optimizer has up-to-date information about the distribution of data values within tables; it cannot be used to compact a clustered index.
2. **Correct Answer: D**
- A. **Incorrect:** The Rebuild Index task does not provide information to the query optimizer.

- B. **Incorrect:** The Reorganize Index task is used to defragment and compact indexes; it does not provide information to the query optimizer.
 - C. **Incorrect:** The Check Database Integrity task is used to locate errors in the database; it does not provide the query optimizer with information.
 - D. **Correct:** The Update Statistics task is used to ensure that the query optimizer has up-to-date information about the distribution of data values within tables.
3. **Correct Answer: A**
- A. **Correct:** SSIS packages are executed using SQL Server agent jobs.
 - B. **Incorrect:** The Clean Up History task is used to remove stale information from the database. It is not related to scheduling execution of custom SSIS packages.
 - C. **Incorrect:** The Update Statistics task is used to provide current information to the query optimizer. It is not related to scheduling execution of custom SSIS packages.
 - D. **Incorrect:** The Check Database Integrity task is used to locate errors within the database. It is not related to scheduling execution of custom SSIS packages.
4. **Correct Answer: A**
- A. **Correct:** Querying the *sys.dm_io_backup_tapes* dynamic management view will provide information about backup tape devices installed on the computer that hosts the SQL Server 2005 instance.
 - B. **Incorrect:** Querying the *sys.dm_io_pending_io_requests* dynamic management view will provide information about pending I/O requests. It will not display information about the backup tape devices.
 - C. **Incorrect:** Querying the *sys.dm_io_shared_drives* dynamic management view will provide information about shared drives in SQL Server 2005 clusters. It will not display information about the backup tape devices.
 - D. **Incorrect:** Querying the *sys.dm_io_virtual_file_stats* dynamic management view will provide information about I/O stats for data and log files. It will not display information about the backup tape devices.

Lesson 3

1. Correct Answers: A and B

- A. **Correct:** Users assigned the Content Manager role are able to perform the Create Linked Reports task.
- B. **Correct:** Users assigned the Publisher role are able to perform the Create Linked Reports task.
- C. **Incorrect:** Users assigned the Browser role are unable to create linked reports. Of the options presented, only the Content Manager and Publisher roles are able to perform the Create Linked Reports task.
- D. **Incorrect:** Users assigned the Report Builder role are unable to create linked reports. Of the options presented, only the Content Manager and Publisher roles are able to perform the Create Linked Reports task.

2. Correct Answer: A

- A. **Correct:** Users assigned the Publisher role are able to create, modify, and delete resources outside the My Reports folder.
- B. **Incorrect:** Users assigned the System User role are unable to create, modify, or delete resources. Users assigned the Publisher and Content Manager roles can perform these tasks.
- C. **Incorrect:** Users who are assigned the My Reports role are able to create, modify, and delete resources within the My Reports folder, but they are unable to do so outside this folder. Users assigned the Publisher and Content Manager roles can perform these tasks.
- D. **Incorrect:** Users assigned the Browser role are unable to create, modify, or delete resources. Users assigned the Publisher and Content Manager roles can perform these tasks.

3. Correct Answer: C

- A. **Incorrect:** The correct URL to access Report Manager is *localhost/Reports*.
- B. **Incorrect:** The correct URL to access Report Manager is *localhost/Reports*.
- C. **Correct:** The correct URL to access Report Manager is *localhost/Reports*.
- D. **Incorrect:** The correct URL to access Report Manager is *localhost/Reports*.

Lesson 4

1. Correct Answers: A and C

- A. **Correct:** To support self-mapping, both the connecting and destination instances must be configured to support Windows authentication.
- B. **Incorrect:** SQL Server authentication does not support self-mapping.
- C. **Correct:** To support self-mapping, both the connecting and destination instances must be configured to support Windows authentication.
- D. **Incorrect:** SQL Server authentication does not support self-mapping.

2. Correct Answer: D

- A. **Incorrect:** Because the other three accounts have no problems with self-mapping, both the connecting and destination instances must support self-mapping.
- B. **Incorrect:** Because the other three accounts have no problems with self-mapping, both the connecting and destination instances must support self-mapping.
- C. **Incorrect:** Self-mapping will still work if the Account Is Trusted For Delegation account property is set within Active Directory.
- D. **Correct:** Accounts that have the Account Is Sensitive And Cannot Be Delegated property set in Active Directory are unable to use self-mapping and must have a special mapping created using the *sp_addlinkedsevrlogin* stored procedure.

Chapter 6: Case Scenario Answers

Case Scenario 1: Managing Updates

- 1. Automatic updates should be disabled on production computers that run SQL Server because updates should be thoroughly tested before being manually applied to mission-critical servers.
- 2. Windows Server Update Services (WSUS) should be placed on the company's screened subnet.
- 3. Microsoft Baseline Security Analyzer can be used to check which updates, hotfixes, and service packs have been applied.

Case Scenario 2: Configuring Report Server Roles

1. The Publisher role will allow Don to perform his duties without allowing him to manage subscriptions.
2. The Report Builder role will allow Jay to perform his duties without granting excess rights.
3. The Content Manager role will allow Darren to publish reports and manage subscriptions.

Chapter 7: Lesson Review Answers

Lesson 1

1. **Correct Answer: A**
 - A. **Correct:** Adding a data flow task makes the Data Flow pane available for adding functionality to transform data.
 - B. **Incorrect:** A File System task is used to manipulate files and folders.
 - C. **Incorrect:** FTP tasks are used to upload and download files from remote computers.
 - D. **Incorrect:** The Send Mail task is used to report information by using e-mail.
2. **Correct Answer: C**
 - A. **Incorrect:** A green line connecting the two tasks indicates that the second task executes only if the first task executes successfully.
 - B. **Incorrect:** A blue line connecting the two tasks indicates that the second task executes when the first task completes its execution.
 - C. **Correct:** A red line connecting the two tasks indicates that the second task executes only if the first task fails in its execution.

Lesson 2

1. **Correct Answer: D**
 - A. **Incorrect:** SSIS packages can be saved to the msdb database.
 - B. **Incorrect:** SSIS package configurations can be saved to an appropriately configured table within any SQL Server database.
 - C. **Incorrect:** SSIS package log files can be saved to a SQL Server database.

- D. **Correct:** SSIS package checkpoint files cannot be saved within any SQL Server database and can be saved only to the file system.

2. **Correct Answers: B and D**

- A. **Incorrect:** Encrypting sensitive information with a user key means that only Ian can access the package's sensitive information. If the package is set using this configuration and Orin opens it, SQL Server replaces the sensitive information with blanks.
- B. **Correct:** Because Ian and Orin have agreed on a shared password, this method could be used, as it protects the sensitive data while allowing them both access.
- C. **Incorrect:** If this method is used, Orin would be unable to open the package at all.
- D. **Correct:** Because Ian and Orin have agreed on a shared password, this method could be used because it encrypts the entire package while allowing them both access.

3. **Correct Answer: D**

- A. **Incorrect:** Although encrypting the package with a user's key almost certainly stops others from editing it, there is no way to ensure that an encrypted package hasn't been changed if no digital signature exists.
- B. **Incorrect:** Although storing the package within SQL Server's msdb database is a good way to secure it, there is no way to ascertain whether the stored package has been altered if no digital signature exists.
- C. **Incorrect:** Although encrypting the package with a password is likely to stop others from editing it, there is no way to ensure that an encrypted package hasn't been changed if no digital signature exists.
- D. **Correct:** If the package is digitally signed at creation, it is possible for a check to occur prior to package execution to see whether the package is the same as the one for which a digital signature exists.

4. **Correct Answers: A and C**

- A. **Correct:** When correctly assigned, the db_dtsadmin role is able to export all packages.
- B. **Incorrect:** The db_dtsltduser role does not have the requisite permissions to export all packages.

- C. **Correct:** When correctly assigned, the `db_dtsoperator` role is able to export all packages.

Lesson 3

1. Correct Answer: A

- A. **Correct:** You must set the *FailPackageOnFailure* property to *True* for each container within the package that you want to set as a restart point.
- B. **Incorrect:** You must set the *FailPackageOnFailure* property to *True* rather than *False* for each container within the package that you want to set as a restart point.
- C. **Incorrect:** The *SaveCheckpoints* property must be set to *True* for it to be possible for a package to be restarted at the point of failure.
- D. **Incorrect:** Changing the setting of the *CheckpointUsage* property will not change how the package executes.

2. Correct Answers: C and D

- A. **Incorrect:** The Supported setting means that a transaction will not be started but will be joined if the parent container has initiated one.
- B. **Incorrect:** The Supported setting means that a transaction will not be started, but will be joined if the parent container has initiated one.
- C. **Correct:** The parent container will not be a part of the transaction.
- D. **Correct:** The child container will initiate a transaction.

3. Correct Answers: A, B, C, and D

- A. **Correct:** It is possible to log Foreach Loop containers within a package even when the package itself is not enabled for logging.
- B. **Correct:** It is possible to log control flow tasks within a package even when the package itself is not enabled for logging.
- C. **Correct:** It is possible to log data flow tasks within a package even when the package itself is not enabled for logging.
- D. **Correct:** It is possible to log File System tasks within a package even when the package itself is not enabled for logging.

Lesson 4

1. **Correct Answers: A and D**
 - A. **Correct:** A properly configured XML file can store multiple package configurations.
 - B. **Incorrect:** Each environment variable can store only one package configuration.
 - C. **Incorrect:** Each registry key can store only one package configuration.
 - D. **Correct:** A properly configured SQL Server table can store multiple package configurations.
2. **Correct Answer: D**
 - A. **Incorrect:** Support files should be placed in the Miscellaneous folder rather than in the Data Sources folder.
 - B. **Incorrect:** Support files should be placed in the Miscellaneous folder rather than in the Data Source Views folder.
 - C. **Incorrect:** Support files should be placed in the Miscellaneous folder rather than in the SSIS Packages folder.
 - D. **Correct:** Support files should be placed in the Miscellaneous folder.
3. **Correct Answer: C**
 - A. **Incorrect:** SSIS packages are not scheduled for execution using BIDS.
 - B. **Incorrect:** SSIS packages are not scheduled for execution using the Package Installation Wizard.
 - C. **Correct:** SSIS packages are scheduled for execution using the SQL Server Agent.
 - D. **Incorrect:** SSIS packages are not scheduled for execution using the Scheduled Tasks utility.
4. **Correct Answer: C**
 - A. **Incorrect:** Add/Remove programs in Control Panel is not used to install SSIS packages.
 - B. **Incorrect:** SQL Server Management Studio is not directly used to install SSIS packages.
 - C. **Correct:** The Package Installation Wizard is used to install SSIS packages.
 - D. **Incorrect:** BIDS is not used to install SSIS packages.

Chapter 7: Case Scenario Answers

Case Scenario 1: Creating and Managing SSIS Packages

1. The File System task is used to move or copy files from one location to another. The FTP task is used to transfer files over the Internet to remote servers. The Send Mail task can also be used.
2. Checkpoints can be used to ensure that only those aspects of the package that failed are reexecuted rather than the entire package.
3. Given that they have Windows Administrator privileges, the exploration vehicle's staff could check the Windows Event Log for problems with package execution. It is possible to get SSIS packages to write events to the Windows Event Log by using the Windows Event Log provider.

Case Scenario 2: SSIS Package Administration

1. A Windows administrator with no SQL Server 2005 privileges can schedule package execution using the combination of a batch file, the Scheduled Tasks utility in the Control Panel, and the dtexec utility.
2. You can digitally sign packages to ensure that they aren't modified after the developers complete work on them.
3. The three methods you could use to copy packages are as follows: creating a deployment utility and copying the files to the SQL Server 2005 Enterprise Edition computers in the production domain, using the Import And Export functionality in SQL Server Management Studio, and using the dtutil utility to copy and move the packages.

Chapter 8: Lesson Review Answers

Lesson 1

1. **Correct Answer: C**
 - A. **Incorrect:** If the Keep The Publisher Change policy is in effect, the Publisher overrides the Subscriber when there is a conflict.
 - B. **Incorrect:** If the Reinitialize The Subscription policy is in effect, the Publisher overrides the Subscriber when there is a conflict.

- C. **Correct:** If the Keep The Subscriber Change policy is in effect, the Subscriber always overrides the Publisher.
2. **Correct Answer: C**
- A. **Incorrect:** SQL Server Integration Services is used to run database packages. SSIS does not detect conflicts.
- B. **Incorrect:** Although SQL Server Agent performs many functions, it is the Queue Reader Agent that detects data conflicts.
- C. **Correct:** The Queue Reader Agent is the SQL Server 2005 subsystem that is responsible for detecting data conflicts during synchronization.
- D. **Incorrect:** SQL Server Analysis Services is used for online analytical processing and data mining. It is not used to detect data conflicts during synchronization.
3. **Correct Answers: A and B**
- A. **Correct:** If a conflict occurred, Ian's change would have been overwritten if the Keep The Publisher Change policy was in effect.
- B. **Correct:** If a conflict occurred, Ian's change would have been overwritten if the Reinitialize The Subscription policy was in effect.
- C. **Incorrect:** If the Keep The Subscriber Change policy was in effect, Ian's change would not have been overwritten on the Publisher if a conflict arose.

Lesson 2

1. **Correct Answer: D**
- A. **Incorrect:** A UNIQUE constraint can enforce uniqueness, but it can't stop users from entering values over a certain limit.
- B. **Incorrect:** A FOREIGN KEY constraint checks values against those that exist in another table. You would use a CHECK constraint to achieve the goal mentioned in this question.
- C. **Incorrect:** A PRIMARY KEY constraint can enforce uniqueness, but it can't stop users from entering values over a certain limit.
- D. **Correct:** A CHECK constraint could be configured to stop users from entering values over 60 into the Weekly_Hours column of the Worksheet table.

2. Correct Answer: C

- A. **Incorrect:** Nullability determines whether a NULL value can be assigned to a column. It does not provide a default value.
- B. **Incorrect:** A CHECK constraint allows or disallows values depending on a Boolean rule. It does not provide a default value.
- C. **Correct:** DEFAULT definitions can apply default values for columns if none are entered directly.
- D. **Incorrect:** A FOREIGN KEY constraint limits the values that can be entered depending on the contents of another column. It does not provide a default value.

3. Correct Answer: A

- A. **Correct:** A FOREIGN KEY constraint targeting the Contractors table could be used on the Contractor_Name column of the Worksheet table to achieve this goal.
- B. **Incorrect:** A UNIQUE constraint would not achieve this goal.
- C. **Incorrect:** You would not use an INSTEAD OF trigger to achieve this goal because this technology is used to execute an alternative statement if a target action is about to occur.
- D. **Incorrect:** You would not use an AFTER trigger to achieve this goal because this technology is used to execute a statement when a target action occurs.

Lesson 3

1. Correct Answer: A

- A. **Correct:** An alias data type can be used to enforce consistency of data precision across a database.
- B. **Incorrect:** You would not use the varchar data type to ensure that measurements are recorded with the same degree of precision.
- C. **Incorrect:** You would not use the char data type to ensure that measurements are recorded with the same degree of precision.
- D. **Incorrect:** You would not use the nchar data type to ensure that measurements are recorded with the same degree of precision.

2. **Correct Answers: B and C**

- A. **Incorrect:** You would store postal codes by using one of the Transact-SQL data types rather than a UDT.
- B. **Correct:** UDTs are suited to storing extremely complex data types, such as geospatial and encrypted data.
- C. **Correct:** UDTs are suited to storing extremely complex data types, such as geospatial and encrypted data.
- D. **Incorrect:** You would store birthday data using one of the Transact-SQL data types rather than a UDT.

Chapter 8: Case Scenario Answers

Case Scenario 1: Making Implicit Constraints Explicit

1. Use the Keep The Publisher Change conflict resolution policy.
2. Use a FOREIGN KEY constraint.
3. Use a CHECK constraint.

Case Scenario 2: Data Types

1. The tinyint data type is most suitable. IPv4 decimal quads can only have a value of 0 to 255.
2. The char or nchar data type would be the most suitable. You would limit the length of either type to 12 characters.
3. You would use the XML data type.

Chapter 9: Lesson Review Answers

Lesson 1

1. **Correct Answer: D**

- A. **Incorrect:** The UNIQUE constraint enforces the uniqueness of the values in a set of columns. It does not ensure that column data meets specific entry requirements.
- B. **Incorrect:** The UPDATE statement changes existing data in a table or view. It does not ensure that column data meets specific entry requirements.

- C. **Incorrect:** The HAVING clause, typically used with a GROUP BY statement, determines the output of a SQL Server 2005 query. It does not ensure that column data meets specific entry requirements.
 - D. **Correct:** You can use the CHECK constraint to ensure that column data entered from any source meets the requirements of the written company policy.
2. **Correct Answer: C**
- A. **Incorrect:** Typically, you use SSMS for managing databases rather than for transforming data.
 - B. **Incorrect:** Typically, you use BIDS for database development rather than for transforming data.
 - C. **Correct:** You use SSIS for extracting, transforming, and loading data.
 - D. **Incorrect:** Typically, you use SSRS for ad hoc and managed reporting rather than for transforming data.
3. **Correct Answer: A**
- A. **Correct:** The SSIS deployment utility can deploy multiple packages, including the package dependencies (for example, configurations and table names). The manifest file lists the packages, the package configurations, and any miscellaneous files in the project.
 - B. **Incorrect:** The UNION statement combines the results of two or more queries into a single result set. It cannot combine data from tables that use different table names.
 - C. **Incorrect:** Backup and restore is not the recommended mechanism for deploying SSIS packages in a production environment. You cannot use the msdb database to merge data from tables that have different table names.
 - D. **Incorrect:** The ALTER TABLE Transact-SQL statement cannot alter table names.
4. **Correct Answers: B, C, and D**
- A. **Incorrect:** The Fuzzy Grouping transformation, not the Fuzzy Lookup transformation, identifies rows of data that are likely to be duplicates and selects a row of data (known as the canonical row) that it uses to standardize the data.

- B. **Correct:** The Fuzzy Lookup transformation provides a default set of delimiters used to tokenize the data. Tokenization is important because it defines the units within the data that are compared to each other.
 - C. **Correct:** The Fuzzy Lookup transformation specifies the maximum number of matches to return per input row.
 - D. **Correct:** The Fuzzy Lookup transformation uses similarity thresholds to derive similarity and confidence scores. Similarity thresholds can be set at the component and join levels.
5. **Correct Answer: C**
- A. **Incorrect:** The manifest file lists the packages, the package configurations, and any miscellaneous files in the project. It does not contain timing information.
 - B. **Incorrect:** The SSIS package has been working fine and suddenly it takes a long time to run. You want to find out why. Redeploying the package could reduce the run time (but probably will not). It does not give you any indication of why the problem occurred.
 - C. **Correct:** The counters associated with the SQL Server:SSIS Pipeline performance object can determine whether any memory issues affect the performance of SSIS packages.
 - D. **Incorrect:** Profiler can create traces that specify event classes in the Security Audit Event Category to help you to verify that users with the appropriate permissions are observing database change control procedures. This does not affect memory issues related to running SSIS packages.

Lesson 2

1. **Correct Answer: B**
- A. **Incorrect:** The UNIQUE constraint ensures that a SELECT statement does not return duplicate rows in a result set. This is not what is required in this scenario.
 - B. **Correct:** The NOT EXISTS subquery, used in a WHERE clause in a stored procedure, can return a list of products that do not have a valid product ID.
 - C. **Incorrect:** A cross join returns the Cartesian product of two tables. This is not what is required in this scenario.

- D. **Incorrect:** The UNION operator returns the result of two queries in a single result set. This is not what is required in this scenario.
2. **Correct Answer: D**
- A. **Incorrect:** A CHECK constraint ensures that data in a column is correctly formatted or is within maximum and minimum values. It does not check whether a numeric value has been increased by more than 5 percent.
- B. **Incorrect:** DDL triggers fire (for example) when a query uses an ALTER TABLE, an ADD TABLE, or a DROP TABLE to change the database structure. Typically, you do not use such statements to increase the value in a column by a percentage.
- C. **Incorrect:** The Audit Database Management event class occurs when a database is created, altered, or dropped. You trace this event class to monitor such events. This is not what is required in this scenario.
- D. **Correct:** Typically, you increase the value in a column by a certain percentage by using the UPDATE statement. DML triggers fire in response to UPDATE, INSERT, or DELETE statements.
3. **Correct Answer: A**
- A. **Correct:** You can create a migration script to convert the structure of the production database to match that of the test database. This automates the procedure and transfers all the changes in a single operation.
- B. **Incorrect:** If you delete the production database and replace it with the test database, the data in the replaced database will be lost.
- C. **Incorrect:** Full outer joins return rows from tables. They cannot be used to merge databases.
- D. **Incorrect:** If you use SSMS, you need to manually add and modify the database objects by using the SSMS user interface. For large databases, this involves excessive administrative effort.

Chapter 9: Case Scenario Answers

Case Scenario 1: Checking and Correcting Invalid Database Entries

1. You should create a DDL trigger that fires on CREATE, ALTER, and DROP statements (for example, ALTER TABLE or DROP DATABASE). The trigger should roll back the transaction. DDL triggers are enabled by default. The trigger

should be disabled only when a migration script runs. You cannot use some statements—for example, CREATE DATABASE—in a DDL trigger. You should instead use event notifications for these statements.

2. You should use the CHECK constraint, which ensures that entries to a column meet specified format requirements.
3. You should use a full outer join on the EmployeeID column. This returns results even when no match exists between records in that column. You can use a WHERE clause to return results on a per-project basis, or to return all results where no match exists on the joining column.
4. You should use the Fuzzy Lookup transformation, which performs data-cleaning tasks such as standardizing data, correcting data, and providing missing values.
5. You should add Don's account to the db_datareader fixed database role.

Case Scenario 2: Managing Schema Changes

1. The DBA team is responsible for implementing schema changes to the pre-production database. The DBA team is also responsible for migrating tested and approved schema changes to the production database.
2. The DBA team holds meetings with the developers to discuss requirements and application specifications. When all parties agree the schema design, the DBA team creates documentation that describes the proposed schema changes. The developers then generate official documentation requesting the schema changes.
3. The developers are responsible for checking and verifying the changes. Until and unless the DBA team receives this verification, either by e-mail or a signed document, schema changes are considered to be temporary.
4. When developers want schema changes propagated to the production database, they need to send a communication to the DBA team by means of a signed form or an e-mail requesting the change. Only approved and tested schema changes can be propagated to the production database.
5. All requests and verifications at every stage of the project need to be recorded, either by means of signed documentation or e-mail. You are responsible for ensuring a paper trail exists that documents every stage of the process.

Chapter 10: Lesson Review Answers

Lesson 1

1. **Correct Answer: C**

- A. **Incorrect:** You specify snapshot replication when information is updated infrequently and it is resource-effective to send a whole new snapshot rather than replicate changes. Snapshot replication is not appropriate for regular, two-way replication.
- B. **Incorrect:** You specify transactional replication when changes to a publication need to be replicated immediately to Subscribers, typically with a latency of two minutes or less. You can configure transactional replication to accept updates at the Subscriber, but more typically you would specify it for one-way replication.
- C. **Correct:** You specify merge replication when replication can be scheduled and does not need to be real-time. Merge replication is typically two-way.
- D. **Incorrect:** You specify the Remote Distributor model if a number of Subscribers exist at a distant location and the remote Distributor receives updates from the Publisher and distributes them to its local Subscribers. This situation does not apply to this scenario.

2. **Correct Answer: B**

- A. **Incorrect:** The constraint exists at the Publisher and cannot be configured on the Subscriber.
- B. **Correct:** To prevent the Subscriber from replicating the CHECK constraint, you need to configure the CHECK constraint with the NOT FOR REPLICATION clause on the table in the Publisher database.
- C. **Incorrect:** The constraint exists at the Publisher and cannot be configured on the Subscriber. Also, a UNIQUE constraint ensures that no duplicate values are entered in specific columns that do not participate in a primary key, which is not what the question requires.
- D. **Incorrect:** A UNIQUE constraint ensures that no duplicate values are entered in specific columns that do not participate in a primary key. This is not relevant in this scenario.

3. **Correct Answer: D**

- A. **Incorrect:** The solution needs to work if all connected disks fail. This includes the quorum disk, so clustering does not meet the specification.

- B. **Incorrect:** Transactional replication provides a degree of fault tolerance, but failover is not its primary purpose. Failover would not be seamless or transparent to users connected to the database when the server fails, and administrator intervention would typically be required to allow database updates on the Subscriber if the Publisher failed.
 - C. **Incorrect:** Log shipping requires that you back up the transaction log on one computer and restore it to the other. If all disks fail on the primary computer, the most recent transactions might be lost. Failover might not, therefore, be seamless. Microsoft recommends the use of database mirroring rather than log shipping.
 - D. **Correct:** Database mirroring is the only solution listed that meets the specification and is therefore the correct answer.
-

Exam Tip Always answer the question asked. In practice, clustering is often the solution to failover requirements. Modern storage solutions using disk arrays very seldom fail completely. However, Question 3 requires the solution to work even if all disks fail. This requirement determines the correct answer.

4. **Correct Answer: B**

- A. **Incorrect:** A warning that a merge length for LAN connections has exceeded or met its threshold tells you whether a merge length exceeds the threshold you specify. Having a merge length that exceeds a specified threshold does not prevent the merge from occurring until the subscription has expired.
- B. **Correct:** A warning that a subscription will expire within the threshold indicates imminent subscription expiration, which will result in the failure of an update.
- C. **Incorrect:** A warning that the rows merged per second for LAN connections has fallen below or met its threshold tells you whether the merge process is meeting the target you specified. Failure to meet this target could indicate a problem, but this does not prevent the merge from occurring until the subscription has expired.
- D. **Incorrect:** The Warn If Latency Exceeds The Threshold warning is used for transactional replication, not merge replication.

5. **Correct Answer: D**

- A. **Incorrect:** The NOT FOR REPLICATION clause stops the trigger from being executed when a replication agent modifies the table on which the

trigger is configured. It does not prevent the trigger from being published as a part of SQL Server replication.

- B. **Incorrect:** The ALL SERVER clause applies the scope of a DDL trigger to the current server. It does not prevent the trigger from being published as a part of SQL Server replication.
 - C. **Incorrect:** The INSTEAD OF clause specifies that the trigger is executed instead of the triggering SQL statement. It does not prevent the trigger from being published as a part of SQL Server replication.
 - D. **Correct:** The WITH ENCRYPTION clause prevents the trigger from being published as part of SQL Server replication.
6. **Correct Answer: A**
- A. **Correct:** Transactional replication replicates data in real time. Push subscriptions push data to the central server whenever the data changes.
 - B. **Incorrect:** Snapshot replication is typically used when changes occur at infrequent intervals but a change involves a large amount of data. Pull subscriptions cause the central Subscriber to pull the data at regular intervals. Data is not sent as soon as it changes.
 - C. **Incorrect:** Merge replication is typically used when replication is two-way and is scheduled rather than occurring in real time. Pull subscriptions cause the central Subscriber to pull the data at regular intervals. Data is not sent as soon as it changes.
 - D. **Incorrect:** Transactional replication replicates data in real time. However, pull subscriptions cause the central Subscriber to pull the data at regular intervals. Data is not sent as soon as it changes.
7. **Correct Answer: C**
- A. **Incorrect:** The Warn If A Subscription Will Expire Within The Threshold warning indicates imminent subscription expiration, resulting in the failure of an update. It does not measure replication latency for all Subscribers.
 - B. **Incorrect:** The Warn If Rows Merged Per Second For LAN Connections Is Less Than The Threshold warning is configured for merge replication, not transactional replication.
 - C. **Correct:** You can use tracer tokens to measure latency either by accessing the Tracer Tokens tab in Replication Monitor or by using Transact-SQL stored procedures.

- D. **Incorrect:** The Warn If Latency Exceeds The Threshold warning indicates that the latency for a particular replication exceeds a predefined limit. It does not measure replication latency for all Subscribers.

Chapter 10: Case Scenario Answers

Case Scenario 1: Selecting Replication Type and Model

1. Northwind Traders should use transactional replication. Depending on the resources available on the SQL Server 2005 server that holds the database table, the company should use either the Central Publisher or Central Publisher with Remote Distributor model. In addition, the Publishing Subscriber model is appropriate for each branch office. This configuration reduces the amount of data that needs to travel over WAN links, and it probably reduces latency.
2. Tailspin Toys should create a snapshot publication and publish this to its stores. The SQL Server 2005 server at the central office is probably configured as a Central Subscriber for merge replication, but this does not prevent it from acting as a Central Publisher for snapshot replication.
3. Humongous Insurance should configure the SQL Server 2005 server at its central office as a Central Subscriber for merge replication. The Central Subscriber collates the information and can then republish the collated information to servers at the branch offices. Humongous Insurance should reconfigure the trigger to use the NOT FOR REPLICATION clause.

Case Scenario 2: Tuning Replication

1. The Publisher can achieve better performance if you move the operating system to a disk where it is not contending with database operation. Performance should improve, and disaster recovery to the point of failure is enabled, if you move the transaction logs to a different disk or disk array than the one hosting the databases. You should also consider placing the publication database on a separate disk or disk array.
2. Transact-SQL stored procedures can use tracer tokens to measure latency. The `sys.sp_posttracertoken` stored procedure posts the tracer token, the `sys.sp_helptracertokens` stored procedure calculates latency, and the `sys.sp_helptracertokenhistory` stored procedure displays tracer token history. You can configure the Warn If Latency Exceeds The Threshold warning in Replication Monitor.

3. You can reduce contention between user activity and replication agent activity by setting the `READ_COMMITTED_SNAPSHOT` database option to `ON` for the publication and subscription databases. When this option is enabled, transactions specifying the `READ_COMMITTED` isolation level (the default) use row versioning instead of locking. When a transaction runs all statements, you will see a snapshot of data as it exists at the start of the statement.

Chapter 11: Lesson Review Answers

Lesson 1

1. **Correct Answer: D**
 - A. **Incorrect:** This password is too short.
 - B. **Incorrect:** This password is the same as the user name. Also, it contains only lowercase and uppercase alphanumeric characters.
 - C. **Incorrect:** This password uses only numeric and lowercase alphanumeric characters.
 - D. **Correct:** This password meets all the criteria. Long, complex passwords are typically used for service accounts.
2. **Correct Answer: A**
 - A. **Correct:** You open SQL Server Management Studio and connect to the appropriate server instance. Object Explorer is available in the left-hand pane of SQL Server Management Studio.
 - B. **Incorrect:** SQL Server Surface Area Configuration lets you configure the surface area by enabling or disabling protocols, services, and features. It does not provide access to Object Explorer.
 - C. **Incorrect:** SQL Server Configuration Manager lets you manage the services and configure the network protocols associated with SQL Server, and to manage the network connectivity configuration from SQL Server client computers. It does not provide access to Object Explorer.
 - D. **Incorrect:** SQL Server Profiler enables you to determine how SQL Server resolves queries internally. It does not provide access to Object Explorer.
3. **Correct Answer: B**
 - A. **Incorrect:** If you place all the SQL Server member servers in a group, it is easier to assign Windows permissions to these servers. However, if the servers

are not in an OU, the only way you can enable the database administrators to grant Windows permissions is to give them this power over the entire domain. This violates the principle of least privilege.

- B. **Correct:** Placing the servers in an OU enables you to delegate permissions for that OU without granting domain-wide permissions.
- C. **Incorrect:** Membership of the sysadmin group on a SQL Server member server enables a user to administer that server. However, not all database administrators have the same level of rights, and they should not all be added to the sysadmin role. This violates the principle of least privilege.
- D. **Incorrect:** MSCS clusters provide failover support. They are not used for delegating administrative privileges.

4. **Correct Answers: A, C, and D**

- A. **Correct:** RSA_512, RSA_1024, and RSA_2048 encryption algorithms can be specified for asymmetric keys.
- B. **Incorrect:** DES and TRIPLE_DES encryption algorithms can be specified for symmetric keys, but not for asymmetric keys.
- C. **Correct:** RSA_1024 is a valid choice. RSA_512, RSA_1024, and RSA_2048 encryption algorithms can be specified for asymmetric keys.
- D. **Correct:** RSA_2048 is a valid choice. RSA_512, RSA_1024, and RSA_2048 encryption algorithms can be specified for asymmetric keys.
- E. **Incorrect:** DES and TRIPLE_DES encryption algorithms can be specified for symmetric keys, but not for asymmetric keys.

5. **Correct Answers: B and E**

- A. **Incorrect:** A certificate authority server issues certificates used for authentication and encryption. It is not concerned with service packs or security updates.
- B. **Correct:** WSUS servers can be configured hierarchically and can obtain service packs or security updates from each other. However, at least one server in the hierarchy needs to connect to the Microsoft Update Web site.
- C. **Incorrect:** VeriSign is a third-party certificate authority. It is not concerned with service packs or security updates.
- D. **Incorrect:** The Windows Update Web site has been superseded by the Microsoft Update Web site.

- E. **Correct:** Service packs and security updates are downloaded from the Microsoft Update Web site.
6. **Correct Answer: C**
- A. **Incorrect:** TCP/IP is the standard protocol for remote connections. Also, this is not the main reason for selecting a protocol.
- B. **Incorrect:** Having more than one connection protocol enabled does not necessarily speed up communication. However, having a protocol enabled when it does not need to be increases the attack surface area of the server.
- C. **Correct:** TCP/IP requires fewer firewall ports to be open and therefore exposes a smaller surface area to attack. The question gives no indication that other machines on your network cannot use TCP/IP, which is the standard network protocol, so no requirement for named pipes exists.
- D. **Incorrect:** Typically, named pipes is neither more efficient nor faster than TCP/IP. This is not, in any case, the main deciding factor. TCP/IP exposes a smaller surface area to attack.

Lesson 2

1. **Correct Answers: A and C**
- A. **Correct:** RSoP analyzes the result of reconfiguring Windows permissions. You can start the RSoP wizard from either the Active Directory Users And Computers or the Active Directory Sites And Services MMC.
- B. **Incorrect:** Object Explorer allows you to verify SQL Server permissions explicitly applied or inherited as a result of role membership. It does not analyze the result of reconfiguring Windows permissions.
- C. **Correct:** RSoP analyzes the result of reconfiguring Windows permissions. You can start the RSoP wizard from either the Active Directory Users And Computers or the Active Directory Sites And Services MMC.
- D. **Incorrect:** Object Explorer allows you to verify SQL Server permissions explicitly applied or inherited as a result of role membership. It does not analyze the result of reconfiguring Windows permissions.
2. **Correct Answers: B and C**
- A. **Incorrect:** The *sp_helpsrvrolemember* stored procedure displays a list of the members of a fixed server role. It does not return information about the password policy settings of a SQL Server login.

- B. **Correct:** The built-in Transact-SQL LOGINPROPERTY function returns information about the password policy settings of a SQL Server login.
 - C. **Correct:** You can use the sys.sql_logins catalog view to determine whether password policy or password expiration is enforced.
 - D. **Incorrect:** The sys.server_permissions Transact-SQL server-level security catalog view returns one row for each server-level permission. You can use it to determine (for example) the identity of the securable on which the permission exists, the permission name and state, and the server principal to which the permission is granted. It does not return information about the password policy settings of a SQL Server login.
3. **Correct Answer: D**
- A. **Incorrect:** Members of the processadmin fixed server role can terminate processes that are running in an instance of SQL Server. This is not what you want Kim to do.
 - B. **Incorrect:** Members of the serveradmin fixed server role can change server-wide configuration options and shut down the server. This is not what you want Kim to do.
 - C. **Incorrect:** Members of the sysadmin fixed server role can perform any activity in the server. This gives Kim more permissions than she needs to perform the specified tasks and hence violates the principle of least privilege.
 - D. **Correct:** Members of the securityadmin fixed server role manage logins and their properties. They can grant, deny, and revoke server-level permissions. They can also grant, deny, and revoke database-level permissions, and reset passwords for SQL Server logins.

Chapter 11: Case Scenario Answers

Case Scenario 1: Configuring Security on SQL Server 2005 Member Servers

1. You should specify Windows integrated authentication for your SQL Server member servers. The credentials for domain user login are then used for SQL Server login, and the user is not required to supply additional credentials. A possible problem with password policy setting is that you do not specify it. The domain administrator team therefore needs to be aware of the requirements of SQL Server login policy.

2. You should ask the domain administrators to create an OU and put the SQL Server member servers in that OU. They should block inheritance on that OU (domain policies with no-override would still apply) and create a GPO to apply group policy to that OU. They should delegate control of that OU to you and your team.
3. You have no permissions to configure domain policy, but domain policies affect your SQL Server member servers. This situation is improved by placing the member servers in an OU, but changes to domainwide policies could still affect your servers. If you have a preproduction network to which you have full administrative rights, you can mirror changes to the production network and test their implications. You can run tools such as RSoP. In particular, you can download service packs and security patches and ensure that these have no adverse effects. Your job description states that you need to advise the domain administrators about any updates that could affect the SQL Server servers.

Case Scenario 2: Adding Your Team Members' User Accounts to Database Roles

1. You should add your team members' accounts to the following database roles:
 - ❑ David: db_datareader, db_datawriter, db_accessadmin
 - ❑ Don: db_datareader, db_datawriter, db_accessadmin, db_backupoperator
 - ❑ Carol: db_datareader, db_datawriter, db_backupoperator
 - ❑ Matt: db_datareader, db_datawriter, db_securityadmin
2. You should add Don's account to the db_ddladmin database role.

Chapter 12: Lesson Review Answers

Lesson 1

1. **Correct Answer: D**
 - A. **Incorrect:** Server room doors should be secured using two-factor authentication. A video surveillance camera does not provide any authentication.
 - B. **Incorrect:** Server room doors should be secured using two-factor authentication. A smartcard reader provides only single-factor authentication.
 - C. **Incorrect:** Server room doors should be secured using two-factor authentication. A fingerprint scanner provides only single-factor authentication.

- D. **Correct:** An RFID tag reader and a PIN number keypad provide two independent ways of verifying identity. This is the correct choice because two-factor authentication should be used to secure the door to a server room.

2. **Correct Answer: A**

- A. **Correct:** Because you are unsure which device is faulty, you need to check both of them against the video footage.
- B. **Incorrect:** Checking the audit log from the fingerprint reader against the video footage will tell you whether the fingerprint reader is faulty, but it won't tell you whether the magnetic card reader is faulty.
- C. **Incorrect:** Checking the audit log from the magnetic swipe card reader against the video footage will tell you whether the magnetic swipe card reader is faulty, but it won't tell you whether the fingerprint reader is faulty.
- D. **Incorrect:** If you check the audit log from the magnetic swipe card reader against the audit log from the fingerprint reader and they each show a different identity, you won't know which audit log is correct unless you check them against the video footage.

3. **Correct Answers: A and B**

- A. **Correct:** Tapes should be rotated to an offsite facility so that if the building burns down or a natural disaster knocks the building out of service, the organization's data can be restored. The facility should be available 24 hours a day, 7 days a week so that you can access the tapes and restore from them as needed.
- B. **Correct:** Tapes should be stored in a fireproof safe. Tapes need to be secured because if they are not, anyone with access to them will be able to access the data they contain.
- C. **Incorrect:** A filing cabinet is not part of a secure backup tape strategy. A filing cabinet is not fireproof and usually isn't able to be well secured.
- D. **Incorrect:** Although a bank safe deposit box would qualify as offsite storage, access would be limited to the hours the bank was open. If, for example, you wanted to perform a restore on the weekend, you'd be out of luck.

4. **Correct Answers: A and D**

- A. **Correct:** Where possible, SQL Server 2005 should be installed on a stand-alone or member server rather than on a domain controller.

- B. **Incorrect:** Where possible, SQL Server 2005 should be installed on a stand-alone or member server rather than on a domain controller.
- C. **Incorrect:** Where possible, configure each SQL Server service to run under a separate, rather than the same, Windows account. That way, if one account is compromised or disabled, only one service is affected.
- D. **Correct:** Where possible, configure each SQL Server service to run under a separate, rather than the same, Windows account. That way, if one account is compromised or disabled, only one service is affected.

Lesson 2

1. Correct Answers: A, B, and C

- A. **Correct:** Barry might know the passwords of some of his colleagues. Even if he does not, getting everyone to change to a new password will ensure that Barry can't get access using another person's credentials.
- B. **Correct:** Although Barry probably doesn't know the password for the application security context account credentials, just to be safe, these should be altered to ensure that Barry can't use them to gain access.
- C. **Correct:** Disabling Barry's account is the first step to take in ensuring that Barry no longer has access to the SQL Server database.
- D. **Incorrect:** Rebooting the computer running SQL Server 2005 will make no difference to any access that Barry has.

2. Correct Answer: B

- A. **Incorrect:** Emergency Management Services provide text interface access to a server that is not in a state to respond to other input. EMS can be used when a server's network stack is unavailable or the server has suffered a STOP error. Unless special hardware is installed, EMS cannot usually be accessed over the network and hence cannot be used in this situation to gracefully shut down SQL Server.
- B. **Correct:** A dedicated administrator connection (DAC) provides a robust way of remotely administering a SQL Server if more traditional remote administration technologies fail.
- C. **Incorrect:** Traditional remote administration technologies usually rely on the RPC protocol to work. Because traditional technologies are not working in this situation, this answer is incorrect.

- D. **Incorrect:** Remote Desktop Protocol is a standard remote administration technology that might cease to function on servers subjected to a DoS attack.
3. **Correct Answers: B and D**
- A. **Incorrect:** Rebooting the computer is unlikely to remove any infection.
- B. **Correct:** Infected computers should be removed from the network so that they do not infect other computers.
- C. **Incorrect:** A DAC is generally used only when the server will not respond to other remote administration technologies. It is unlikely to be used to connect to an infected server that should have already been removed from the network.
- D. **Correct:** Antivirus vendors release new definitions for their products on a regular basis. An update might have been released since the server last automatically updated its definitions. Attempt to update the antivirus definitions, and perform a full scan of the server.

Chapter 12: Case Scenario Answers

Case Scenario 1: Physically Securing a Server Room

1. Because this room was used as a storeroom rather than as a server room, quite likely it has sprinklers installed to stop fires from spreading. These sprinklers need to be removed and replaced by a halon fire suppression system so that any fire that occurs in the room can be extinguished without destroying the servers.
2. The design elements you need to incorporate into the door locking system are two-factor authentication, the ability to log each person's entry into the room, and the ability to automatically lock the door if power is lost. For safety reasons, when the door automatically locks people inside should be able to exit. In such a situation, their entry is already logged.
3. You need to check that the room's roof is securely reinforced so that someone can't get up on a stepladder outside the room, lift a few ceiling tiles, and crawl over into the server room.

Case Scenario 2: Responding to a Denial of Service Attack

1. You can configure Performance Monitor alerts to note unusual increases in traffic. A sharp rise in traffic or logins might indicate a DoS attack is in progress.
2. You should make an attempt to shut down SQL Server using a DAC (dedicated administrator connection).

System Requirements

We recommend that you use an isolated network that is not part of your production network to do the practice exercises in this book. You need a two-station network that you can implement either by using two computers connected by a crossover network cable or by using a single computer running virtual machine software.

Hardware Requirements

Your computer or computers should meet the following hardware specification:

- Personal computer with a 600-MHz Pentium III-compatible or faster processor (Pentium IV or equivalent if you plan to use virtual machine software)
- 512 MB of RAM (1.5 GB if you plan to use virtual machine software)
- 10 GB of available hard disk space (20 GB if you plan to use virtual machine software)
- DVD-ROM drive
- Super VGA (1024 x 768) or higher resolution video adapter and monitor
- Keyboard and Microsoft mouse, or compatible pointing device

Software Requirements

The following software is required to complete the practice exercises:

- Microsoft Windows 2003 Server with Service Pack 2 (SP2) or later
- Microsoft SQL Server 2005 Enterprise Edition, SP1 or later (A 180-day evaluation edition of Microsoft SQL Server 2005 Enterprise Edition is included on the DVD that comes with this book.)
- The latest version of the AdventureWorks database (which you can find at www.microsoft.com/downloads)

MORE INFO Software requirements

For more details about these software requirements, please see the Appendix.

IMPORTANT Evaluation edition is not the full retail product

The 180-day evaluation edition of Microsoft SQL Server 2005 Enterprise Edition provided with this training kit is not the full retail product and is provided only for the purposes of training and evaluation. Microsoft and Microsoft Technical Support do not support this evaluation edition.

Information about any issues relating to the use of this evaluation edition with this training kit is posted to the Support section of the Microsoft Press Web site (www.microsoft.com/learning/support/books/). For information about ordering the full version of any Microsoft software, please call Microsoft Sales at (800) 426-9400 or visit www.microsoft.com.

About the Authors

Orin Thomas

Orin Thomas, MCSE, CCNA, CCDA, and Linux+ certified, is an author and systems and database administrator. He is the convener of the Melbourne Infrastructure Administrator's group, the coauthor of several Training Kits for Microsoft Press, and a contributing editor for *Windows IT Pro* magazine.

Ian McLean

Ian McLean, MCSE, MCITP, MCT, has 40 years experience in industry, commerce, and education. He started his career as an electronics engineer before going into distance learning and then education as a university professor. Currently he runs his own consultancy company. Ian has written 18 books plus many papers and technical articles. He has been working with SQL since it was SEQUEL and has qualifications in Microsoft SQL Server 6.0, 6.5, 7.0, 2000, and 2005.