Ross Mistry

with Hilary Cotter

Based on **Windows Server 2008**

Microsoft® SQL Server 2008

Management and Administration

SAMS

Microsoft® SQL Server 2008 Management and Administration

ISBN-13: 978-0-672-33044-5
ISBN-10: 0-672-33044-X

Library of Congress Cataloging-in-Publication Data

Mistry, Ross.

Microsoft SQL server 2008 management and administration / Ross Mistry.

– 1st ed.

p. cm.

ISBN 978-0-672-33044-5

1. SQL server. 2. Database management. I. Title.

QA76.9.D3M57886 2008

005.4'476–dc22

2008048922

Printed in the United States of America

First Printing December 2009

Editor-in-Chief
Karen Gettman

Executive Editor
Neil Rowe

Development Editor
Mark Renfrow

Managing Editor
Patrick Kanouse

Project Editor
Mandie Frank

Copy Editor
Margaret Berson

Indexer
Ken Johnson

Proofreader
Matt Purcell

Technical Editor
Todd Robinson, MCITP

Publishing Coordinator
Cindy Teeters

Designer
Gary Adair

Compositor
TnT Design, Inc.

Contributing Writers
Hilary Cotter, MVP;
Shirmattie Seenarine;
John Welch, MVP;
Marco Shaw, MVP;
Maciej Pilecki, MVP

INTRODUCTION

SQL Server 2008 is Microsoft's latest data platform providing data management and analytical solutions for the enterprise. The platform is trusted, ensures business continuity, and is more predictable and more scalable than ever before. Although similar to SQL Server 2005, SQL Server 2008 offers a tremendous number of new features and functionality for database administrators, developers, and business intelligence architects.

This book is designed to be the ultimate guide for database administrators as it provides detailed guidance in the areas of planning, installation, management, administration, security, high availability, monitoring, and performance tuning of a SQL Server environment. Moreover, this book includes industry best practices, tips, and step-by-step instructions based on real-world examples.

Some of the classic and new SQL Server 2008 topics covered in the book include: installation, upgrade and migration strategies, Policy Based Administration, Resource Governor, encryption, failover clustering, database mirroring, authorization, hardening, consolidation and virtualization, maintenance plans, monitoring, performance tuning, troubleshooting, log shipping, PowerShell scripting, replication, creating packages and transferring data, indexes, full-text catalogs, and backing up and restoring databases.

The book is also based on Microsoft's latest award-winning server operating system—Windows Server 2008. As a result, not only will readers gain knowledge about SQL Server 2008, but they will also have the opportunity to understand the advantages of running SQL Server 2008 on Windows Server 2008.

What Is in This Book?

This book is organized into five parts, with each part made up of several chapters focusing on core SQL Server 2008 elements. The parts and chapters of the book are detailed in this section.

Part I: Installing, Administering, and Managing the Database Engine

The first part of the book begins by providing an overview of SQL Server 2008, including planning and installing the new platform. After you get SQL Server 2008 installed, the majority of your time will be spent managing and administering the new SQL Server infrastructure. Therefore, the remainder of Part I consists of chapters dedicated to SQL Server 2008 administration and management tasks.

Chapter 1: Installing or Upgrading to the SQL Server 2008 Database Engine

With the recent release of SQL Server 2008, organizations are eager to migrate to the new and improved database platform. However, many organizations feel challenged when trying to establish the best strategies for moving forward. This chapter focuses on the various SQL Server 2008 migration strategies that are available. It answers the question once and for all whether or not organizations should upgrade from a previous version or perform a new SQL Server 2008 installation from scratch and then conduct a migration.

Other topics highlighted in this chapter include: supported migration methodologies, hardware requirements, using the SQL Server 2008 Upgrade Advisor, supported legacy versions of SQL Server, and best practices, tips, and common pitfalls to look out for to achieve a successful migration. Moreover, this chapter describes the benefits associated with running SQL Server on Windows Server 2008, and it also includes upgrade strategies for moving to the latest server operating system.

Chapter 2: Administering the SQL Server 2008 Database Engine

After SQL Server 2008 is installed, it is necessary to configure and administer the server. This chapter focuses on administering the core features and components associated with the Database Engine. Topics include administering the SQL Server properties pages, Database properties pages, Database Engine folders, and the SQL Server Agent. Managing server and database configuration settings—such as memory, processor performance, auditing, compression, database files, and autogrowth—is also covered in depth.

Chapter 3: Creating Packages and Transferring Data with Integration Services

A common database administrator task is transferring data or databases between source and target environments. This chapter focuses on importing, exporting, and transforming data and databases via SQL Server Management Studio and Integration Services. The chapter also covers how packages are created, saved, and executed as well as the management of the Integration Services component.

Chapter 4: Managing and Optimizing SQL Server 2008 Indexes

Similar to an index found in a book, an index in SQL Server is utilized for fast retrieval of data from tables. This chapter explains index concepts, ways to design the appropriate index strategy to maximize performance, creating indexes with SQL Server Management Studio, and how to create indexes with Transact-SQL. The chapter also introduces new SQL Server 2008 index topics, such as creating spatial and filtered indexes, and shares best practices on implementing, managing, and optimizing indexes.

Chapter 5: Administering SQL Server 2008 Full-Text Search

With the data explosion and the ever-increasing amount of data being stored in its native format, full-text search is playing an increasingly important role in databases today. This chapter discusses the new features in SQL Server full-text search and provides step-by-step instructions on how to implement full-text search on your tables, and best practices for full-text search.

Chapter 6: SQL Server 2008 Maintenance Practices

For SQL Server to perform at optimal levels, a DBA should conduct routine maintenance on each database. This chapter focuses on best practices associated with maintaining databases within the SQL Server Database Engine.

The discussion includes creating maintenance plans to check database integrity, shrink databases, reorganize indexes, and update statistics. Additionally, this chapter provides recommendations on daily, weekly, monthly, and quarterly maintenance practices to be conducted on SQL Servers.

Chapter 7: Backing Up and Restoring the SQL Server 2008 Database Engine

Backing and restoring databases is one of the most critical duties of a DBA. It is important that the DBA understand the concepts associated with SQL Server backups, therefore, in the event of a disaster they can restore the database to the point of failure. This chapter covers the new backup compression feature, the importance of backups, creating a backup and recovery plan, storage architecture, transaction log files, recovery model, the various types of backups, backing up with SSMS, automating backups with a maintenance plan, backing up full-text catalogs, creating database snapshots, and of course, best practices.

Part II: SQL Server 2008 Security Practices

Part II of SQL Server 2008 Management and Administration is dedicated to SQL Server security. The first two chapters cover hardening techniques for both SQL Server 2008 and Windows Server 2008 and also discuss administering security and authorization. The next two chapters describe the new features, Policy Based Management, and encryption.

Chapter 8: Hardening a SQL Server Implementation

SQL Server is regularly targeted by hackers because it is a repository of sensitive data for organizations. If an organization's system is breached, hackers can gain access to confidential information including, but not limited to, credit card numbers, social security numbers, and marketing information. As such, it is imperative that database administrators secure both the SQL Server implementation and the data residing in it. This chapter provides an

overview of how to harden a SQL Server implementation based on industry best practices so that vulnerabilities and security breaches are minimized.

Some of the security and hardening topics that the chapter delves into include: choosing the appropriate authentication mechanism, hardening the SA account, enforcing strong passwords, leveraging the configuration tools to lock down a SQL Server, configuring the Windows Server 2008 advanced firewall for secure access, selecting the correct service account, and applying security templates with Active Directory.

Chapter 9: Administering SQL Server Security and Authorization

After the SQL Server installation is hardened, the next step involves administering security and granting authorization to the SQL Server environment. Chapter 9 is all about security administration topics: creating logons, granting access and authorization, understanding SQL Server roles, administering password policies, endpoint authentication, SQL Server and database principals, role-based security, and user and schema separation.

Chapter 10: Administering Policy Based Management

Enforcing best practices and standardization on large SQL Server installations was extremely difficult in previous versions of SQL Server. To ensure standardization, SQL Server 2008 introduces Policy Based Management, which allows a DBA to define policies that can be applied to one or more SQL Server instances, databases, and objects. Policy Based Management works on SQL Server 2000 and all its successors.

New concepts, components, terminology, and reporting with Policy Based Management as well as best practices are discussed in Chapter 10. The reader's attention is then turned to real-world step-by-step examples of how to implement Policy Based Management in their environment.

Chapter 11: Encrypting SQL Server Data and Communications

Organizations and DBAs are facing excessive pressure from regulatory agencies to ensure that mission-critical data stored within SQL Server is encrypted. Not only is it important to ensure that data stored within SQL Server is encrypted, but it is equally important to ensure that data in transit is encrypted, database encryption can be implemented without making changes to an application, data can be encrypted at the cell level, and finally, SQL Server volumes can also be encrypted.

Chapter 11 shares many strategies, tips, and best practices on how to leverage the encryption-based technologies included in SQL Server 2008 and Windows Server 2008 for end-to-end data protection. Among these strategies are: leveraging transparent data encryption, integrating security with a

Hardware Security Module (HSM), using certificates to encrypt data in transit, and encrypting SQL Server volumes with BitLocker.

Part III: SQL Server 2008 High-Availability Alternatives

DBAs typically feel compelled to choose just the right technologies to achieve high availability when designing a SQL Server infrastructure. Some of the challenges they face are illustrated in their questions: Should I use failover clustering, database mirroring, log shipping, or replication? Which alternative provides the best protection? How does Windows Server 2008 impact my decision? This part of the book will alleviate pressure and concerns by providing DBAs with best practices and tips on how to design and choose the right SQL Server 2008 high-availability alternative to meet their organization's needs.

In this part, the chapters aim to teach DBAs how to select the appropriate HA technology when designing and implementing SQL Server 2008. Technologies include failover clustering, log shipping, peer-to-peer replication, and database mirroring, based on Windows Server 2008.

Chapter 12: Implementing and Managing Failover Clustering

The main focus of Chapter 12 is on how to design, configure, and manage a single-instance or multiple-instance high-availability failover cluster with SQL Server 2008.

The chapter is also based on Windows Server 2008. As a result, it includes the new features and prerequisites associated with implementing failover clustering with SQL Server 2008 on Windows Server 2008. Step-by-step procedures for both the Windows Server 2008 failover cluster and SQL Server 2008 failover cluster are also provided.

Chapter 13: Implementing and Managing Database Mirroring

Chapter 13 deals with configuring and managing database mirroring so that organizations can enhance the availability of their SQL Server databases, increase business continuity, and maintain a hot standby of their database in another geographic location. The chapter includes detailed step-by-step instructions for configurations of all three database mirroring modes: high availability, high protection, and high performance.

This chapter also includes best practices from the field, case studies, and discussions of how to integrate database mirroring with other high-availability alternatives such as failover clustering and how to recover from a failed server.

Chapter 14: Implementing and Managing Log Shipping

The focus of Chapter 14 is on configuring and managing log shipping. This chapter supplies step-by-step instructions on how to configure and maintain

one or more warm standby databases, typically referred to as *secondary databases*. Like the other high-availability chapters in this part, this chapter includes real-world examples and industry best practices.

Chapter 15: Implementing and Managing SQL Server Replication

SQL Server Replication is another way of distributing data from a source SQL Server to either one or more target SQL Servers. This chapter focuses on replication components and provides a prelude to the different types of replication scenarios that a database administrator can manage, such as snapshot, merge, and transactional replication. Step-by-step replication configurations, including the peer-to-peer replication scenario, a new form of high availability, are also presented.

Part IV: Monitoring and Troubleshooting SQL Server 2008

Part IV of this book covers monitoring, troubleshooting, and performance-tuning techniques to ensure that SQL Server 2008 is optimized and performing at the highest possible levels.

Chapter 16: Managing Workloads and Consumption with Resource Governor

Unfortunately, with the previous releases of SQL Server, there was no way to effectively mitigate performance issues associated with SQL Server workloads. Thankfully, with SQL Server 2008, Microsoft introduced Resource Governor, which provides persistent performance to end users and applications by giving Database Administrators the potential to define resource limits and priorities on different SQL Server workloads. This chapter focuses on introducing, implementing, and configuring Resource Governor from a database administrator perspective.

Chapter 17: Monitoring SQL Server 2008 with Native Tools

SQL Server 2008 includes a tremendous number of native tools that should be leveraged in order to monitor a SQL Server database and infrastructure. This chapter first teaches a DBA how to use native tools to gain quick insight into a SQL Server system. It then focuses on how to leverage the new SQL Server audit functionality to monitor events and SQL Server activity and how to configure database mail. The final technologies described in this chapter are Performance Studio, and Performance and Reliability Monitoring, which are new monitoring technologies introduced with SQL Server 2008 and Windows Server 2008.

Chapter 18: Proactively Monitoring SQL Server 2008 with System Center Operations Manager 2007

Database Administrators and IT managers are under constant pressure to ensure that their SQL Server systems are operating at optimal levels. This can be very

challenging when managing a large SQL Server infrastructure. It is recommended for DBAs to leverage Operations Manager 2007 to proactively monitor and provide a comprehensive view of the health of a SQL Server infrastructure.

The discussion in this chapter focuses on the Operations Manager components, installing the SQL Server management pack, and on how to use the operations console to effectively monitor a SQL Server infrastructure. By gaining the upper hand in monitoring and managing a SQL Server infrastructure, Database Administrators can achieve the agility necessary to be successful and focus on other tasks.

Chapter 19: Performance Tuning and Troubleshooting SQL Server 2008

After SQL Server 2008 is placed in operation and is being monitored, it is important to take action on the findings. It is often difficult to anticipate real-world loads during the development phase of application deployment; thus, it is critical to adjust the parameters of the SQL Server 2008 platform to optimize the performance after it is deployed. Frequently, DBAs need to troubleshoot the performance of SQL Server 2008 to address problems that are uncovered by monitoring.

This chapter focuses on how to tune, optimize, and troubleshoot the performance of the SQL Server 2008 system. Specific tools and components include Windows Server 2008 System Monitor, performance thresholds, SQL Server Profiler, Database Engine Tuning Advisor, Query Analysis, and Extended Events.

Part V: SQL Server 2008 Bonus Content

The final part of the book offers bonus content, including leveraging PowerShell to administer SQL Server and implementing Hyper-V to virtualize a SQL Server infrastructure.

Chapter 20: Administering SQL Server 2008 with PowerShell

SQL Server 2008 introduces support for Windows PowerShell. PowerShell is a new command-line shell and scripting language for automating server administration. Microsoft is beginning to integrate this new scripting language into all of their server products. This chapter discusses how to install PowerShell, provides an overview of the language and how it is integrated into SQL 2008, and then dives into how PowerShell can be leveraged to undertake common SQL Server tasks.

Chapter 21: Consolidating and Virtualizing SQL Server 2008 with Hyper-V

Tremendous efforts are being made in the IT industry and the world to sustain the environment. Going green by consolidating and virtualizing servers is a great way to simplify management, minimize data center costs, reduce power

consumption, and make the world a better place for future generations. By consolidating SQL Server instances and leveraging Windows Server 2008 Hyper-V for consolidating and virtualizing SQL Server databases and instances, organizations can reduce the number of SQL Server systems within the infrastructure. This chapter focuses on how to plan and implement Hyper-V and create virtualized guest sessions so that SQL Server systems can be consolidated and virtualized. In addition, SQL Server virtualization considerations are also discussed.

Sample Databases

To facilitate running the steps introduced throughout the book, all of the examples are based on either the AdventureWorks2008 or Customer database. Both of these database samples can be downloaded directly from Microsoft's SQL Server sample website at http://www.codeplex.com/SqlServerSamples. The website and downloads also include the step-by-step instructions on how to install the sample databases on SQL Server 2008.

The exact link to the OLTP Adventureworks2008 sample can be found at http://www.codeplex.com/MSFTDBProdSamples/Release/ProjectReleases.aspx?ReleaseId=18407. Be sure to choose the correct MSI file based on the 2008 schema and the processor type of your system.

The Customer database can be downloaded from the sample Integration Services Product Samples website at http://www.codeplex.com/MSFTISProdSamples. The package sample name is "Execute SQL Statements in a Loop Sample Package" and the customer data which needs to be imported into a newly created Customer database is located in Customer.txt file.

In addition, the book is based on SQL Server 2008 Enterprise Edition running on Windows Server 2008 Enterprise Edition. The following elements were used for the scenarios of the book.

Element	Description
Domain	Companyabc.com
Domain Controller	TOR-DC01.companyabc.com
Main Site	Toronto, Ontario
SQL Server Name	TOR-SQL01\Instance01 TOR-SQL01\Instance02
Secondary Site	San Francisco, California
SQL Server Name	SFC-SQL01\Instance01 SFC-SQL02\Instance02
Disaster Recovery Site	New York, New York NYC-SQL01\Instance01 NYC-SQL02\Instance02

CHAPTER 1

Installing or Upgrading to the SQL Server 2008 Database Engine

SQL Server 2008 Database Engine Services, formerly referred to as "The Database Engine," is the nucleus of SQL Server 2008. Its features are nothing short of impressive. Many organizations today run complex data applications that command the use of a feature like SQL Server's Database Engine that can process, store, and secure data. From a transactional perspective, it is used to store, process, and secure data for the most demanding data-consuming applications within your enterprise. Moreover, the Database Engine offers many other benefits and advantages for organizations. It controls authorization to SQL Server objects, provides high-availability functionality, and includes subfeatures such as Replication and Full-Text Search.

This chapter describes the step-by-step process for installing a clean version of SQL Server 2008 Database Engine and/or upgrading an existing SQL Server Database Engine implementation to SQL Server 2008. In addition, this chapter covers how to leverage the planning tools to ensure a successful installation, upgrade, or transition.

Even though the SQL Server 2008 installation process is very intuitive and has been simplified, a DBA must make several key decisions to ensure that the completed installation or upgrade will meet the needs of the organization. For example, is it beneficial to upgrade an existing SQL Server implementation to SQL Server 2008, or is it preferred to conduct a clean install from scratch? What are the ramifications of these alternatives? Will you lose your existing SQL Server settings, databases, and configurations?

This chapter covers these prerequisite planning tasks to address the questions and concerns of DBAs.

In addition, this chapter also covers the hardware and software prerequisites, supported SQL Server 2008 upgrade paths, supported Windows operating systems for running SQL Server 2008, and benefits of running SQL Server 2008 on the highly anticipated Windows Server 2008 family of operating systems.

What's New for Installation with SQL Server 2008?

SQL Server continues to deliver a robust experience when installing or upgrading to SQL Server 2008. Moreover, SQL Server 2008 introduces significant enhancements to make the installation or upgrade process even more simple and seamless compared to its predecessors. The new installation features for SQL Server 2008 consist of the following:

- A new SQL Server 2008 Installation Center landing page, which includes a tremendous number of options for planning, installing, and maintaining a SQL Server implementation. The Installation Center is also a one-stop shop for planning and reviewing SQL Server documentation before getting started.
- A new planning tool known as the System Configuration Checker, which checks for conditions that could impede a SQL Server installation prior to the actual installation.
- Another great planning tool is the newly refined Install Upgrade Advisor. The Install Upgrade Advisor tool allows a DBA to fully analyze existing SQL Server 2005 and SQL Server 2000 installations for issues that may surface when upgrading to SQL Server 2008. By fixing these issues before conducting the upgrade, an organization will have a smoother experience when transitioning to SQL Server 2008.
- New maintenance tasks have been introduced in the installation process, which allow DBAs to either repair a corrupt SQL Server 2008 installation or conduct a Feature Upgrade. The Feature Upgrade tool is a wizard that allows organizations to upgrade or change their existing edition of SQL Server 2008 after the initial installation or upgrade is complete.
- A discovery report that will provide a detailed report of all SQL Server 2000, 2005, and 2008 components, features, and settings associated with an installation.

- The potential to automate SQL Server installations by using an existing configuration file.
- An Advanced Cluster Preparation tool, which streamlines and prepares a SQL Server 2008 failover cluster installation—which is typically deemed a very difficult task in the industry.
- A tool that will allow for a smooth transition of packages by automatically upgrading them from SQL Server 2005 to the SQL Server 2008 Integration Services format.

Deprecated SQL Server 2008 Database Engine Elements

Not only is it essential to understand the new features and functionality associated with the Database Engine, but it is equally important to understand older elements that have been deprecated in SQL Server 2008. Let's examine the list of deprecated elements that are no longer supported or associated with the Database Engine.

- The `DUMP` statement associated with backups has been deprecated and replaced with `BACKUP`.
- The `LOAD` statement associated with restores has been replaced with `RESTORE`.
- `BACKUP LOG WITH NO_LOG` and `BACKUP LOG WITH TRUNCATE_ONLY` are no longer available as the transaction log is automatically truncated with the database using the Simple recovery model.
- The `sp_addalias` procedure has been removed.
- SQL Server 60, 65, and 70 compatibility levels are no longer supported. Databases must at least maintain a compatibility level of 80.
- The `sp_addgroup`, `sp_dropgroup`, and `sp_helpgroup` stored procedures have been replaced with roles.
- Sample databases such as Northwind, Pubs, and AdventureWorks are no longer included as optional installation features with the installation program. If you want to use the sample databases, they are offered as out-of-band downloads from Microsoft's samples website.
- The Surface Area Configuration (SAC) tool has been replaced with Policy Based Management.
- Remote servers have been replaced with Linked servers.

Note

The preceding bullets itemize the main features deprecated with the Database Engine. However, for a full list of each item, review the topic "Deprecated Database Engine Features and Discontinued Database Engine Functionality in SQL Server 2008" in SQL Server Books Online (BOL).

Preplanning and Preparing a SQL Server 2008 Server Database Engine Installation

Before you begin the actual installation of SQL Server 2008 Database Engine, you must make several decisions concerning preliminary tasks. How well you plan these steps will determine how successful your installation is—as many of these decisions cannot be changed after the installation is complete.

Verifying Minimum Hardware Requirements

Whether you are installing SQL Server 2008 in a lab or production environment, you need to ensure that the hardware chosen meets the minimum system requirements. In most situations, the minimum hardware requirements presented will not suffice; therefore, Table 1.1 provides not only the minimum requirements, but also the recommended and optimal system requirements for the hardware components.

Note

This book is tailored toward the Standard and Enterprise Editions. As such, the minimum hardware and software requirements documented in Table 1.1 and Table 1.2 of this chapter only cover "core editions" of SQL Server 2008, also known as Standard and Enterprise Editions. To review the hardware and software requirements for the "specialized editions," refer to the section on "Hardware and Software Requirements for Installing SQL Server 2008" in SQL Server 2008 Books Online (BOL).

Table 1.1 **SQL Server 2008 Processor and Memory System Requirements**

SQL Server 2008 Enterprise Edition (64-bit) IA64		
Component	**Minimum Requirements**	**Recommended Requirements**
Processor	Itanium	1.0GHz or higher: Itanium
Memory	512MB	2.048GB or above

SQL Server 2008 Enterprise Edition (64-bit) x64		
Component	**Minimum Requirements**	**Recommended Requirements**
Processor	1.4GHz: AMD Opteron, AMD Athlon, Intel Xeon EM64T, and Intel Pentium IV EM64T	2GHz or higher: AMD Opteron, AMD Athlon, Intel Xeon EM64T, and Intel Pentium IV EM64T
Memory	512MB	2.048GB or above
SQL Server 2008 Standard Edition (64-bit) x64		
Component	**Minimum Requirements**	**Recommended Requirements**
Processor	1.4GHz: AMD Opteron, AMD Athlon, Intel Xeon EM64T, and Intel Pentium IV EM64T	2GHz or higher: AMD Opteron, AMD Athlon, Intel Xeon EM64T, and Intel Pentium IV EM64T
Memory	512MB	2.048GB or above
SQL Server 2008 Enterprise Edition (32-bit) x64		
Component	**Minimum Requirements**	**Recommended Requirements**
Processor	1.0GHz: Pentium III	2GHz or higher: Pentium III
Memory	512MB	2.048GB or above
SQL Server 2008 Standard Edition (32-bit)		
Component	**Minimum Requirements**	**Recommended Requirements**
Processor	1.0GHz: Pentium III	2GHz or higher: Pentium III
Memory	512MB	2.048GB or above

The minimum disk space requirements differ depending on which SQL Server 2008 feature will be installed. Table 1.2 depicts these minimum disk space specifications itemized by feature.

Table 1.2 **SQL Server 2008 Minimum Disk Requirements**

SQL Server 2008 Feature	Minimum Disk Space Required in MB
Database Engine and data files, Replication, and Full-Text Search	280
Analysis Services and data files	90
Reporting Services and Report Manager	120
Integration Services	120
Client Components	850
SQL Server Books Online (BOL) and SQL Server Compact Books Online	240

Note

When designing and selecting the system specifications for a SQL Server implementation, even the optimal system requirements recommendations from Microsoft might not suffice. It is a best practice to assess the server specifications of the planned server role while taking the load during the time of deployment and future growth into consideration. For example, a SQL Server 2008 system running 50 instances of the Database Engine will require much more than the recommended specification of 2GB of RAM to run adequately. In addition, SQL Server 2008 running on a Windows 2008 server that is providing business intelligence solutions for 10,000 users might require 32GB of RAM. Therefore, size the system accordingly and test the load before going live into production.

Examining SQL Server 2008 Software Prerequisites

Before installing SQL Server 2008, it is also important to get acquainted with the software prerequisites, as many of these prerequisites outline best practices. As such, you should take the time to review the prerequisites before implementation to ensure installation or upgrade success. The SQL Server 2008 software prerequisites include

- .NET Framework 3.5
- Windows Installer 4.5 or later
- Microsoft Data Access Components (MDAC) 2.8 SP1 or later
- Internet Explorer 6 SP1 or later
- Latest version of PowerShell
- Latest Windows Server hot fixes are recommended
- If SQL Server 2008 will be virtualized, then Hyper-V is required and supported.

Note

For more information and consolidating and virtualizing SQL Server 2008 on Windows Server 2008 Hyper-V, refer to Chapter 21, "Consolidating and Virtualizating with Hyper-V."

The SQL Server installation wizard will first verify if these software prerequisites are already installed. If they are not, don't panic—the SQL Server 2008 installation wizard is very intuitive and will most likely prompt, and then install all of these software prerequisites automatically. Therefore, you

won't have to spend hours conducting Google searches, trying to nail down the appropriate downloads, including versions.

Choosing the Appropriate SQL Server Edition

SQL Server 2008 comes in a variety of editions that are tailored to suit the needs and requirements of different organizations and applications. The SQL Server 2008 Editions include the Enterprise, Standard, Workgroup, Express, Compact, and Developer Editions, as described in the following sections.

SQL Server 2008 Enterprise Edition

The SQL Server 2008 Enterprise Edition is the complete feature set of the product and is designed to support the needs of the largest enterprises. It includes all the features for scalability, performance, high availability, enterprise security, data warehousing, business intelligence, and enterprise manageability. The Enterprise Edition is fully 64-bit capable, is optimized to run on 64-bit platforms and can support all the processors and memory found in the operating system.

Some other new features only found in the Enterprise edition include Partitioned Table Parallelism, enhanced database mirroring features, Resource Governor, Backup Compression, online operations, Hot Add CPU, Performance Data Collector, Extensible Key Management, Failover Clustering, Transparent Data Encryption, and Change Data Capture.

SQL Server 2008 Standard Edition

The SQL Server 2008 Standard Edition includes the core set of functionality needed to support data warehouses, electronic commerce applications, and line-of-business applications. It is designed to support the needs of small to medium organizations. The Standard Edition is fully 64-bit capable and can support a maximum of four processors. It is worth mentioning that two nodes of failover clustering is also supported within the Standard Edition.

SQL Server 2008 Workgroup Edition

The SQL Server 2008 Workgroup Edition is designed for small organizations and includes the core database features needed for applications.

SQL Server 2008 Express Edition

The SQL Server 2008 Express Edition is the free edition that is designed to support small or targeted applications with a core set of secure database requirements. This edition replaces the Microsoft SQL Server Desktop Engine (MSDE) platform available in SQL Server 2000 and augments the Express Edition in SQL Server 2005.

SQL Server 2008 Compact Edition

The SQL Server 2008 Compact Edition is the free edition that runs on mobile devices as well as desktops. This provides a single lightweight database platform for client applications. This edition replaces the SQL Server Mobile product and augments the SQL Server 2005 Compact Edition.

SQL Server 2008 Developer Edition

The SQL Server 2008 Developer Edition provides all the same features and functionality as the Enterprise Edition but is licensed only for development purposes.

The following link includes the full list of features supported based on the Editions of SQL Server 2008.

http://msdn.microsoft.com/en-us/library/cc645993.aspx

Choosing the Appropriate Windows Operating System Version and Edition to Support the SQL Server Installation

SQL Server 2008 can run on a number of Windows operating systems. SQL Server 2008 can run on top of Windows Server 2008, Windows Server 2003 SP2, Windows VISTA, and Windows XP. When referring to Windows Server 2008, either the Windows Server 2008 edition with or without Hyper-V can be utilized. Please note that SQL Server 2008 does not support running Windows 2000 or Windows NT 4.0.

Benefits of Running SQL Server 2008 on Windows Server 2008

Hands down, the Windows Server 2008 family of operating systems is the best choice for running SQL Server 2008. By combining the two products, the highest level of security, scalability, reliability, high availability and compliance can be achieved. Some of the major benefits of running SQL Server 2008 on Windows Server 2008 include the following:

- **Authentication**—The Windows Server 2008 authentication mechanism provides the highest level of security for authorization when running Active Directory Domain Services. SQL Server can leverage the following: Active Directory role-based security for authorization and administration, two-factor authentication with SmartCard-based certificates and biometric devices, and integration with certificate services. Finally, Kerberos is now supported for all SQL Server protocols.
- **Encryption**—By combining the encryption technologies included in both SQL Server 2008 and Windows Server 2008, it is finally possible to achieve encryption from an end-to-end perspective.

- **Minimized Footprint**—Both Windows Server 2008 and SQL Server 2008 provide a modularized installation process that is very granular. Therefore, you only install what you need. This strategy minimizes the attack surface, which in turn, mitigates breaches and compromises.
- **Compliance**—New features and functionality such as integrating Audit and Audit Specifications directly with the Windows Server 2008 event and security logs allows for stronger auditing functionality, which is a requirement of many major regulatory compliances.
- **Dynamic Hardware Partitioning**—Allows for both CPU and RAM to be added to the SQL Server system on the fly, without causing a server outage.
- **High Availability Clustering**—Windows Server 2008 supports up to 16 nodes within a SQL Server 2008 failover cluster. In addition, the requirement of having all nodes within the same subnet has been alleviated. Consequently, with the new quorum model and no subnet restriction, it is easier to achieve geographically dispersed clusters.
- **Policy Based Management**—By leveraging the Windows Server 2008 group policy and the configuration management strategies, policies can be created to manage SQL Server databases settings and configurations.
- **PowerShell**—The latest scripting technology geared toward effectively managing Windows Server and Microsoft applications has extended to SQL Server 2008. DBAs can use the powerful command-line scripting technologies to automate administrator tasks for both Windows Server 2008 and SQL Server 2008.
- **Performance Management**—Windows Server 2008 introduces Windows Reliability and a newly refined performance monitor tool for troubleshooting and monitoring SQL Server system performance. In addition, the Windows performance framework has been augmented through the introduction of SQL Server 2008 Performance Data Collector. As a result, collecting, analyzing, and troubleshooting SQL Server data in a centralized solution for end-to-end monitoring can be achieved.
- **Consolidation and Virtualization**—Hyper-V has been introduced with Windows Server 2008. Hyper-V is Microsoft's virtualization technology. By using Hyper-V in conjunction with SQL Server 2008, you can consolidate SQL Servers into a virtualized environment, reduce SQL Server hardware, and reduce total cost of ownership within the infrastructure.

Understanding the Windows Server 2008 Family of Operating Systems

In the Windows 2008 family of operating systems, there are four main editions, and SQL Server 2008 can run on any of them. These editions include Windows Server 2008 Standard, Windows Server 2008 Enterprise Edition, Windows Server 2008 Datacenter Edition, and Windows Web Server 2008.

Organizations and DBAs must understand their workload needs and requirements when selecting the appropriate Windows Server 2008 operating system edition to utilize. In addition, the Windows Server edition selected must also coincide with requirements pertaining to the edition of SQL Server 2008 selected. For example, the Windows Server 2008 Enterprise Edition might be selected if there is a need to sustain an eight-node SQL Server failover cluster; the Standard Edition may be selected in order to save on licensing costs; or Windows Server 2008 Enterprise Edition with Hyper-V may be selected if there is a need to virtualize the SQL Server environment.

Each edition supports both the 32-bit and 64-bit processor architectures and the Standard, Enterprise, and Datacenter editions ship with or without Hyper-V. Hyper-V is the latest Microsoft virtualization technology based on Windows Server 2008.

Finally, when running SQL Server 2008 on Windows Server 2008, the maximum amount of RAM supported by the operating system on 32-bit systems is 4GB when running the Standard Edition and 64GB when running the Enterprise and Datacenter Editions. For a 64-bit system, the numbers increase as the Standard Edition can support 32GB and the Enterprise and Datacenter Editions can support up to 2TB of RAM. Hence it is strongly recommended to be using the x64 versions of the operating system whenever possible as it allows for greater flexibility and upgradability.

> **Caution**
>
> When installing Windows Server 2008, it is possible to select a Server Core installation. Windows Server 2008 Server Core is a stripped-down minimal installation that provides a low-maintenance environment through limited functionality. The present version of Server Core is not intended to be an application platform. Since there are SQL Server dependencies that are not part of the Server Core, SQL Server 2008 *cannot* run on a Windows Server Core installation.

For an overview of the editions and a complete list of Windows 2008 features and functionality, refer to the Microsoft Windows Server 2008 home page at the following link:
http://www.microsoft.com/windowsserver2008/en/us/default.aspx.

New Installation, Upgrade, or Transition?

Organizations that have conducted a SQL Server implementation in the past may need to perform a new SQL Server 2008 installation, a side-by-side installation or upgrade their existing SQL Server system, which is commonly referred to as an *in-place* upgrade. Finally, organizations may choose to transition to SQL Server 2008 by first installing a new installation and then migrating SQL Server databases and objects from the legacy environment. There are benefits to each of these options. The next two sections detail the benefits.

Should You Perform a New SQL Server 2008 Installation?

The primary benefit of a new installation is that, by installing the operating system from scratch, you are starting with a known good server and a brand new SQL Server 2008 implementation. You can avoid migrating problems that might have existed on your previous server—whether due to corrupt software, incorrect configuration settings, or improperly installed applications. Moreover, a new installation provides an opportunity for housecleaning as legacy SQL Server items are not carried over.

For example, it is common for an old SQL Server system to have many outdated databases, packages, user accounts, and stored procedures that have not been touched in over 10 years. Keep in mind, however, that you will also lose all configuration settings from your previous installation. In addition, all SQL Server elements, such as databases, user accounts, packages, and so on, will need to be migrated/transitioned. Moreover, required applications on the legacy server will need to be reinstalled after the installation of the new operating system and the SQL Server 2008 implementation are complete. Make sure you document your server configuration information and back up any data that you want to keep.

When running SQL Server 2008, there may be situations where installing a new installation from scratch is the only option. For example, it is not possible to upgrade a legacy SQL Server Failover Cluster from SQL Server 2005 running on Windows Server 2003 to SQL Server 2008 Failover Clustering running on Windows Server 2008.

Note

When performing a new installation of SQL Server 2008, it is possible to install a new SQL Server 2008 instance on an existing system with SQL Server 2005. Therefore, a side-by-side installation is supported and a migration of existing data from SQL Server 2005 to SQL Server 2008 can be achieved all within the same server.

Should You Upgrade an Existing SQL Server System to SQL Server 2008?

Upgrading, on the other hand, replaces your current SQL Server binaries but keeps existing databases, components, features, packages, users, settings, groups, rights, and permissions intact. In this scenario, you don't have to reinstall applications or restore data. Before choosing this option, keep in mind that you should test your applications and databases for compatibility before migration. Just because they worked on previous versions of SQL Server does not mean they will work on SQL Server 2008.

As always, before performing any type of server maintenance such as a SQL Server or Windows Server 2008 in-place upgrade, you should perform a complete backup of the SQL Server environment, any applications residing on the server, and data that you want to preserve. Do not forget to include the System State when backing up the SQL Server system. It is required when performing a restore if you want to maintain the existing Windows settings.

Table 1.3 lists the upgrade paths for SQL Server 2008.

Table 1.3 **SQL Server 2008 Upgrade Paths**

Previous SQL Server System	Upgrade to SQL Server 2008
SQL Server 2008	SQL Server version upgrades supported
SQL Server 2005	Yes, fully supported to like edition
SQL Server 2000	Yes, fully supported to like edition
SQL Server 7.0	Not supported
SQL Server 6.5	Not supported

Note

In-place upgrades can only be accomplished when using the same edition. For example, an upgrade from SQL Server 2005 Standard to SQL Server 2008 Enterprise cannot be achieved. Nevertheless, there is a way around this situation. It is possible to upgrade from SQL Server 2005 Standard to SQL Server 2008 Standard and then conduct an edition upgrade to SQL Server 2008 Enterprise.

Should You Upgrade the Operating System to Windows Server 2008?

On another note, when upgrading an existing system to SQL Server 2008, there may be situations when an organization would also want to upgrade the underlying operating system to Windows Server 2008. To upgrade to Windows 2008, you must be running a server-level operating system. You

cannot upgrade Workstation or Home Editions of operating systems such as Windows XP or Windows Vista to Windows 2008. To upgrade your existing SQL Server's operating system, you must be running Windows Server 2003. An upgrade from Windows NT 4.0 and Windows 2000 Server is not permitted. Table 1.4 lists the available upgrade paths to Windows 2008.

Table 1.4 **Windows Server 2008 Upgrade Paths**

Previous Operating System	Upgrade to Windows Server 2008
Microsoft Windows Server 2003 R2 Standard, Enterprise, or Datacenter Edition	Yes, fully supported to like edition
Microsoft Windows Server 2003 operating systems with Service Pack 1 (SP1) Standard, Enterprise, or Datacenter Edition	Yes, fully supported to like edition
Microsoft Windows Server 2003 operating systems with Service Pack 2 (SP2) Standard, Enterprise, or Datacenter Edition	Yes, fully supported to like edition
Windows NT 4.0	Not supported
Windows 2000 Server	Not supported
Windows XP	Not supported
Windows Vista	Not supported

Note

For Windows Server 2008, in-place upgrades can only be performed to the same editions of Windows Server 2008. For example, an upgrade from Windows Server 2003 Standard to Windows Server 2008 Enterprise cannot be achieved.

Gathering Additional Information Necessary to Proceed

During the installation of SQL Server 2008, you will have to tell the setup wizard how you want your server configured. The wizard will take the information you provide and configure the server settings to meet your specifications.

Taking the time to gather the information described in the following sections before starting your installation or upgrade will likely make your SQL Server 2008 installation go faster, smoother, and easier.

New SQL Server 2008 Installation or In-place Upgrade

The first and most major decision when moving toward SQL Server 2008 is debating whether to implement a brand new SQL Server installation from scratch or to conduct an in-place upgrade. If you don't already have SQL Server in your existing infrastructure, it is a "no-brainer," and a new installation is warranted. However, if a legacy version of SQL Server resides in the infrastructure, the organization must decide between an in-place upgrade or a new installation. If a new installation is chosen, it is necessary to transition existing SQL Server data from the legacy system to the newly established SQL Server 2008 system. As mentioned earlier, each alternative has benefits and disadvantages.

New SQL Server 2008 Stand-alone Installation or Failover Cluster

Another major decision needs to be made in the planning phases: Should SQL Server 2008 be installed in a stand-alone system or should Failover Clustering be utilized? Failover Clustering provides high availability for a SQL Server instance and should be leveraged if an organization needs maximum availability, protection against server hardware failure, seamless failover that does not require DBA intervention, and finally, automatic client redirects. A stand-alone installation is also sufficient, as it is cheaper, easier to administer, and does not require specific failover clustering hardware.

> **Note**
>
> To install SQL Server 2008 Failover Cluster, review Chapter 12, "Implementing and Managing Failover Clustering."

Single-Instance or Multiple-Instance Installation

For years now, discussions on the topic of single-instance versus multiple-installation have both engulfed and engaged the SQL Server community. Should you install a single-instance SQL Server installation and place all databases on one instance, or scale up and create a multiple-instance SQL Server installation and spread databases across each of these instances? This question continues to echo through every organization. Here are some best practices to assist in making such an arduous decision.

One of the main drawbacks of placing all databases on a single-instance installation involves the tempdb database. The tempdb database is a shared resource between all databases contained within the same SQL Server

instance. Performance degradation may occur as the tempdb database is the single point of contention for all temporary database workloads. In multiple-instance installations, a tempdb database is created for each instance, minimizing contention and performance degradation.

Many DBAs implement multiple instances for other reasons, including regulatory compliance, administrator autonomy, different global security policies, global server settings, and compatibility requirements.

> **Note**
>
> Only one instance within the installation can maintain the default instance name. Therefore, if a default instance already exists, SQL Server 2008 must be installed as a named instance.

Side-by-Side Installations with Previous Versions of SQL Server

Organizations also have the option to install a brand new installation of SQL Server 2008 on a server that is already running a legacy instances of SQL Server 2005. Based on this methodology, more than one version of SQL Server will reside on the system.

Typically, the preference is to either conduct an in-place upgrade or install SQL Server 2008 on new hardware to minimize hardware contention and performance degradation. However, side-by-side installations are sometimes warranted. Let's look at the situations that support this installation. SQL Server 2008 will coexist with SQL Server 2005 and SQL Server 2000. Unfortunately, SQL Server 7.0 is not supported, but hopefully the majority of the organizations out there have already transitioned out of SQL Server 7.0 because it is no longer supported by Microsoft.

Determine Which SQL Server 2008 Features to Install

Give serious thought to the SQL Server 2008 features before installing them. The modular setup of SQL Server 2008 is made up of many independent features, previously referred to as components, allowing for complete customization by organizations. This typically results in minimal surface area and more granularity compared with older editions of SQL Server. This improved modular installation process is said to be "slim and efficient" like other new Microsoft products such as Windows Server 2008 and Exchange Server 2007.

The following bullets depict the modular installation including shared features that can be selected during the installation of SQL Server 2008:

- **Database Engine Services**—This is the core service for storing, processing, and securing data. It is designed to provide a scalable, fast, and high-availability platform for access and the other components. Two subfeatures within the Database Engine are
  - **SQL Server Replication**—Replication allows DBAs to copy databases to different locations and keep the copies synchronized. This can be used for data distribution, synchronization, fault tolerance, disaster recovery, load balancing, or testing. The Replication component manages database replication and interacts primarily with the Database Engine features.
  - **Full-Text Search**—The Full-Text Search engine populates and manages the full-text catalogs. The Full-Text engine also makes full-text searches easier by maintaining indexes, a thesaurus, noise words, and linguistic analysis of the full-text indexes.
- **Analysis Services**—The SQL Server 2008 Analysis Services (SSAS) feature provides online analytical processing (OLAP) and data mining. OLAP is a modification of the original database concept of online transaction processing (OLTP). OLAP is designed to provide immediate answers to analytical and ad hoc queries from a multidimensional cube known as an OLAP cube. Data mining is the process of searching large volumes of data for patterns and trends. SSAS allows SQL Server 2008 to provide both these capabilities and is the core feature of business intelligence.
- **Reporting Services**—The Microsoft SQL Server 2008 Reporting Services (SSRS) feature allows for the presentation and delivery of data in a variety of ways. The reports can include tables, matrices, and free-form data. The source data for the reports can be provided by the Database Engine component, the Analysis Services component, or any Microsoft .NET data provider such as ODBC or OLE DB to access data sources such as Oracle or file-based data.
- **Shared Features**—Features designated as "Shared Features" include
  - **Business Intelligence Development Studio**—The Business Intelligence Development Studio is essentially Microsoft Visual Studio 2008 with some additional SQL Server 2008 business intelligence project types. It is an applications development environment that allows developers to build applications that include Analysis Services, Integration Services, and Reporting Services.
  - **Client Tools Connectivity**—This feature includes the installation of communication components between clients and servers.

- **Integration Services**—The SQL Server 2008 Integration Services (SSIS) feature integrates data from different sources. This integration includes importing, exporting, and transforming data from disparate sources. The data can be copied, merged, restructured, and cleaned as part of the integration processing, which makes the integration services a powerful tool in the development of data warehouses. It is imperative to mention that the Integration Services component fills an important gap in the extract.
- **Client Tools Backward Compatibility**—This feature was heavily requested by the SQL Server community. When Client Tools Backward Compatibility is installed, a DBA can manage legacy SQL Server systems.
- **Client Tools SDK**—This feature includes the Software Development Kit containing resources for developers.
- **SQL Server Books Online**—SQL Server Books Online (BOL) is Microsoft's documentation for SQL Server 2008.
- **Management Tools Complete**—When installed, SQL Server 2008 will possess all the management tools, including but not limited to Management Studio, support for Reporting Services, Analysis Services, Integration Services, SQL Server Profiler, and Database Tuning Advisor.
- **Management Tools Basic**—This refers to the scaled-down version of the management toolset. It only includes management studio support for the Database Engine, SQL Server Express, SQL Server Command-Line Utility, and PowerShell.
- **SQL Client Connectivity SDK**—This feature includes the Software Development Kit containing connectivity resources for developers.
- **Microsoft Sync Framework**—This is a comprehensive synchronization platform enabling collaboration and offline of applications, services and devices with support for any data type, any data store, any transfer protocol, and network topology.

When installing the SQL Server 2008 Database Engine, the additional optional subfeatures to install include

- Replication
- Full-Text Search

- Integration Services
- Connectivity components
- Programming models
- Management tools
- Management Studio
- SQL Server Books Online

Installing a Clean Version of SQL Server 2008

The following sections depict the step-by-step instructions for installing a clean version of the SQL Server 2008 Database Engine feature including supplementary subfeatures such as SQL Server Replication, Full-Text Search, Integration Services, and Shared Components.

1. Log in to the server with administrative privileges and insert the SQL Server 2008 media. Autorun should launch the SQL Server 2008 Installation Center landing page; otherwise, click Setup.exe

> **Note**
>
> If SQL Server's setup software prerequisites have not been met, the installation wizard will prompt, and then install the prerequisites. After the prerequisites have been installed, the SQL Server installation process will resume. SQL Server 2008 software prerequisites may include hotfixes, .NET Framework 3.5 and the latest Windows Installer, version 4.5. In addition, system restarts may be required after SQL Server's setup software prerequisites are installed. If so, rerun setup after the reboot to continue with the SQL Server installation.

2. On the SQL Server Installation Center landing page, first select the Installation page, and then click the New SQL Server Stand-alone Installation or Add Features to an Existing Installation link, as displayed in Figure 1.1.
3. On the Setup Support Rules page, review the outcome of the System Configuration Checker. Ensure that all tests associated with the operation passed without any failures, warnings, or skipped elements. Alternatively, you can review a standard or comprehensive report by selecting the Show Details button or View Detailed Report. To continue with the installation, click OK, as illustrated in Figure 1.2.

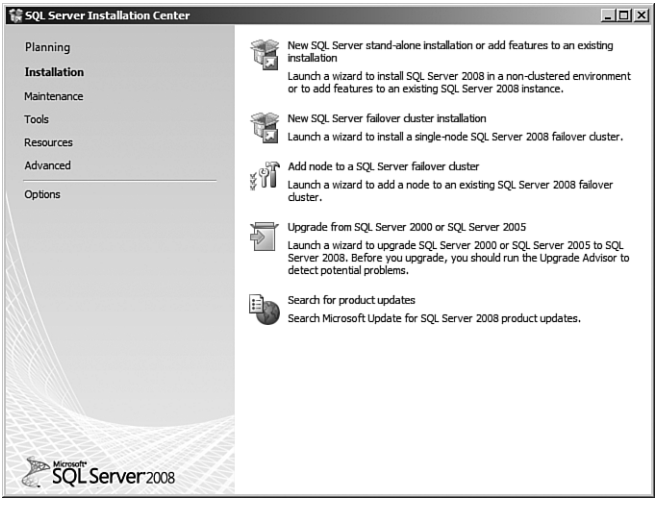

FIGURE 1.1
Performing a new SQL Server stand-alone installation.

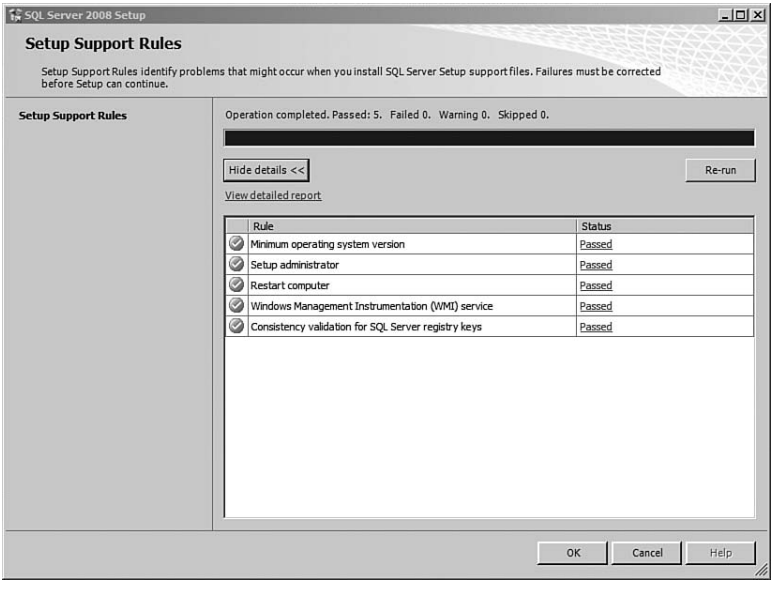

FIGURE 1.2
Reviewing potential problems identified with the Setup Support Rules.

4. On the Product Key page, enter the SQL Server Product Key and click Next.
5. On the License Agreement page, Accept the Licensing Terms, and then click Next.
6. On the Setup Support Files page, the wizard will prompt whether or not additional setup support files are required for the installation. If additional files are required, review the status of the files required, and then click Install.
7. The Setup Support Rules page will be displayed again and will identify any outstanding items that may hinder the installation process associated with the SQL Server installation. Review and correct failures and warnings before commencing the installation. If failures are not displayed, click Next to start the installation. Once any outstanding installation are complete, review the details, and then click Next.

Note

Some of the items that will be tested for in step 7 are: Fusion Template Library, Unsupported SQL Server products, whether the server is a Domain Controller, the version of Windows PowerShell, and Windows Firewall Settings.

8. On the Feature Selection page, select the desired features to be installed and provide the path for the Shared Feature Directory. For this example, the Database Engine Services, SQL Server Replication, Full-Text Search, and appropriate Shared Features such as Integration Services and the Management Tools have been selected. Click Next to proceed as illustrated in Figure 1.3.
9. On the Instance Configuration page, specify the Name and Instance ID for the SQL Server installation. The options include either the Default Instance name, which is MSSQLServer, or a Named Instance. In addition, click the ellipsis button in the Instance Root Directory area and specify the path for the installation. Click Next as displayed in Figure 1.4.

Note

Each instance name provided must be unique and there can only be one default instance per SQL Server system.

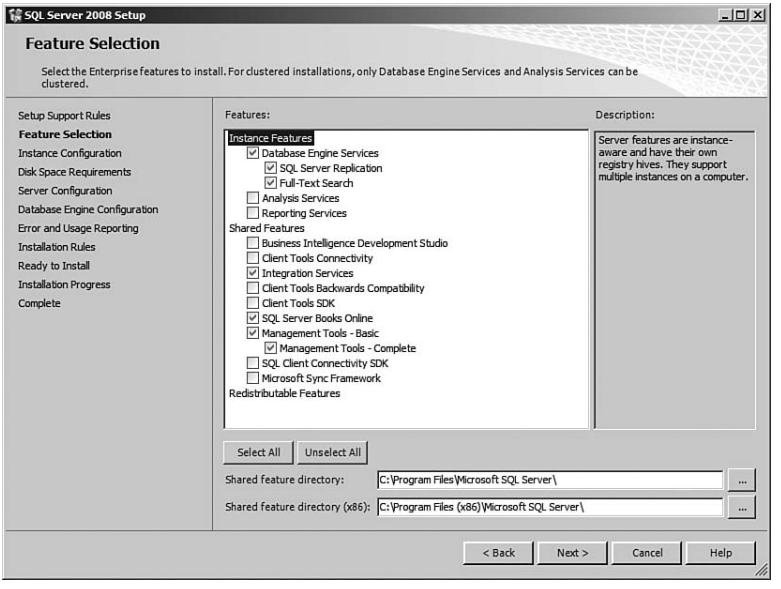

FIGURE 1.3
Specifying the SQL Server features to be installed.

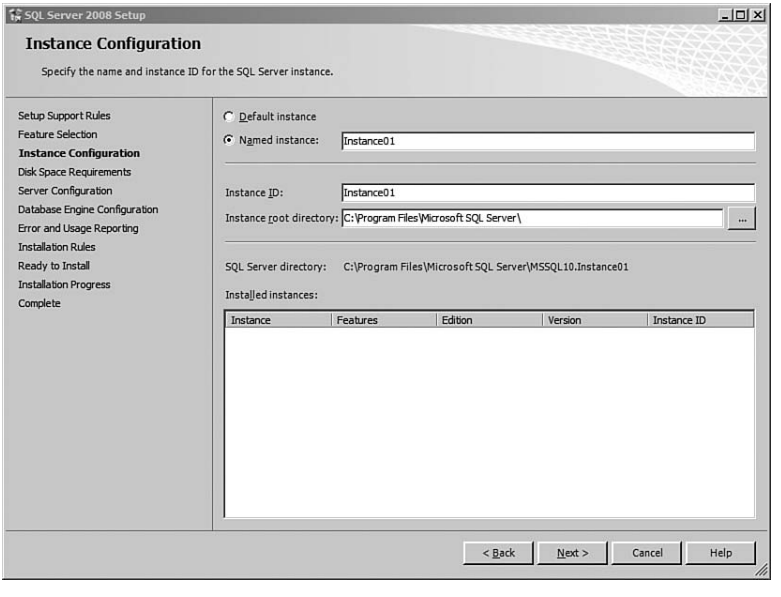

FIGURE 1.4
Configuring the SQL Server instance.

10. The next page is the Disk Space Requirements. Review the disk space summary for the SQL Server components and features selected to be installed, and then click Next.

11. The Server Configuration page includes configuration settings for both Service Accounts and Collation. On the Service Accounts tab, enter a valid low-privilege service account name and password for each service account. Next, specify the Startup Type for each service account listed, as illustrated in Figure 1.5. Options include Automatic, Manual or Disabled. Before proceeding to the next step, click the Collation tab,

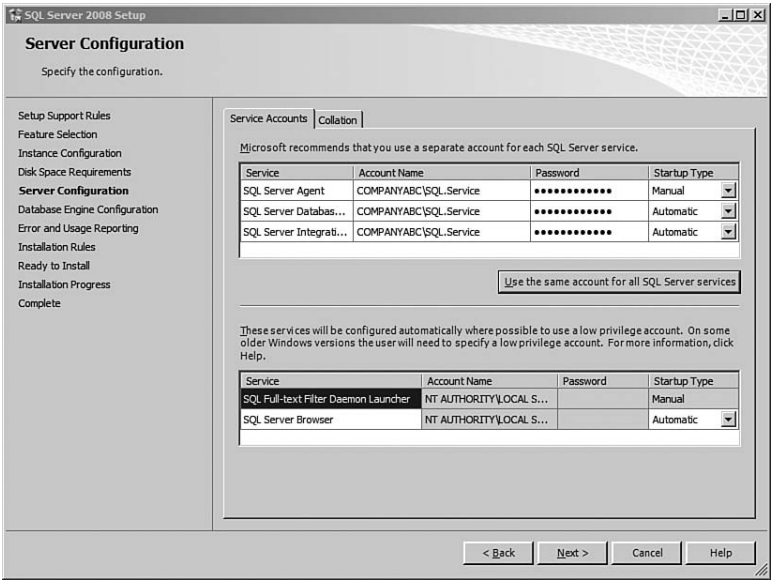

FIGURE 1.5
Specifying the SQL Server service accounts.

Note

From a hardening perspective, Microsoft recommends entering a separate service account for each SQL Server component and feature being installed. In addition, the account specified should follow the principle of least privilege. For more information on selecting the desired service account, and hardening a SQL Server implementation, see Chapter 8, "Hardening a SQL Server Implementation."

12. On the Collation tab, enter the desired collation option for the Database Engine. It is possible to change default collation settings used by the Database Engine and Analysis Services for language and sorting by selecting Customize. Click Next to continue.

13. The Database Engine Configuration page consists of three tabs. The tabs include Account Provisioning, Data Directories, and FILESTREAM. On the first tab, in the Account Provisioning section, specify the Authentication Mode, which consists of either Windows Authentication Mode or Mixed Mode (SQL Server authentication and Windows authentication). If mixed mode is selected, enter and confirm the password for the Built-in SQL Server administrator account. The next step is to provision a SQL Server Administrator by either selecting the option Add Current User or clicking Add and specifying a SQL Server administrator account.

Note

New to SQL Server 2008 is the opportunity to rename the SA account during installation. Renaming the SA account increases security as the account name is well known in the industry.

14. The second tab, Data Directories, located still within the Database Engine Configuration page, is used for specifying the location of the default directories associated with the installation of this SQL Server instance. The directories include Data Root Directory, System Database Directory, User Database Directory, User Database Log Directory, TempDB Directory, TempDB Log Directory, and Backup Directory. Either maintain the default directories or specify a new directory for performance and availability.

Tip

Because I/O to log files is sequential and I/O to database files is random, for increased performance, it is a best practice to place log files on a separate disk from database files. In addition, placing the tempdb on its own disk also bolsters performance.

15. The final tab on the Database Engine Configuration page is FILESTREAM. Here, decide whether you want to enable FILESTREAM. If FILESTREAM is enabled, additional parameters must

be entered such as Enable FILESTREAM for File I/O Streaming Access, Windows Share Name, and whether to allow remote clients to have streaming access to FILESTREAM data. Click Next to proceed.

16. On the Error and Usage Reporting page, help Microsoft improve SQL Server features and services by sending error reports and feature usage to Microsoft. Specify the level of participation, and then click Next.
17. The final check will take place to ensure that the installation process will not be blocked. On the Installation Rules page, review for any outstanding errors or warnings and then click Next to continue.
18. Before commencing the SQL Server 2008 Installation, review the features to be installed on the Ready to Install page, and then click Install.
19. When the installation process starts, you can monitor its progress accordingly. When the installation setup completes, review the success status based on each SQL Server feature and then click Next.
20. On the Complete page, review the location of the SQL Server summary upgrade log file and additional items, which can be found in the supplemental information section. Click Close to finalize the installation.
21. To conduct post-installation tasks, review the upcoming section "Finalizing the SQL Server 2008 Installation or Upgrade" in this chapter.

Upgrading the Database Engine to SQL Server 2008

When upgrading an existing SQL Server system to SQL Server 2008, all SQL Server databases, configuration settings, security settings, and programs are retained from the previous installation. However, there are still several important prerequisite tasks that you perform before the upgrade, as discussed in the following sections.

Tip

It is not possible to change the installation path when upgrading a system to SQL Server 2008. In addition, there must be enough free space on the system and SQL Server partition to support the upgrade; otherwise, the upgrade will come to a halt.

Creating a SQL Server Feature Discovery Report

One of the first tasks a DBA should conduct when upgrading an existing SQL Server system to SQL Server 2008 is to create a discovery report. A SQL Server discovery report, ultimately, is an inventory of the SQL Server components and features installed on an existing SQL Server installation. SQL Server 2008 comes with a new tool called the SQL Server Feature Discovery Report, which will generate a list of features and products. This report can automatically be generated for SQL Server 2008, SQL Server 2005, and SQL Server 2000, by selecting Installed SQL Server Features Discovery Report, on the Tools page, located on the SQL Server Installation Center landing page.

Backing Up the Server

Whenever you are making a major change on a server, something could go wrong. A complete backup of the SQL Server environment, including the SQL Server system databases and Windows Server System State, can make the difference between confidently telling the boss you had a setback so you conducted a rollback, or quivering while you try to find a way to tell your boss a complete disaster has taken place.

Verifying System Compatibility

The first action when upgrading an existing SQL Server system to SQL Server 2008 is running the System Configuration Checker. Launch the System Configuration Checker by first selecting Planning and then System Configuration Checker via the SQL Server 2008 Installation Center landing page.

The System Configuration Checker is an informative tool that conducts a scan on the existing system and indicates problems that might occur when the SQL Server support files are installed. After the scan is completed, a detailed report is presented that indicates the operations that passed, failed, skipped, or presented warnings. View the detailed report, correct any issues, and rerun the scan to ensure absolute success. Then move on to the next prerequisite task, which is running the SQL Server 2008 Upgrade Advisor.

Running the SQL Server Upgrade Advisor

Make it a prerequisite task to test the existing SQL Server system that you plan on upgrading for compatibility issues. Accomplish this by running the SQL Server Upgrade Advisor. The SQL Server Upgrade Advisor is an intuitive tool included with the SQL Server 2008 installation media.

When invoked, the wizard will first analyze previously installed SQL Server components and then produce a detailed report indicating possible upgrade anomalies. In addition, the report provides links to information on how to resolve the issues identified—how convenient!

Caution

Before conducting an in-place upgrade, it is imperative to acknowledge and fix all anomalies. If anomalies go unresolved, the upgrade is sure to fail, resulting in a production outage.

It is important to mention that the Upgrade Advisor can be installed on a remote system and still analyze the following SQL Server components: the Database Engine, Analysis Services, Reporting Services, Integration Services, and Data Transformation Services. Notification Services is not included as this component has been deprecated in SQL Server 2008.

Note

The exhaustive analysis performed by the wizard is unable to examine stored procedures if they are encrypted. Moreover, it is necessary to input a password if DTS or Integration Services packages are password protected.

Installing the SQL Server Upgrade Advisor

1. Launch the SQL Server Installation Center.
2. Select the Planning link and then click Install Upgrade Advisor.
3. Click Next on the SQL Server 2008 Upgrade Advisor Setup Welcome screen.
4. Accept the License Agreement, and then click Next.
5. On the Registration Information screen, enter your name and company name, and then click Next.
6. Provide the installation path on the Feature Selection page. Click Next to continue.
7. Click Install to initiate the installation, and then click Finish to finalize.

Performing a Compatibility Test with SQL Server Upgrade Advisor

When running the SQL Server 2008 Upgrade Advisor, the high-level steps include identifying SQL Server components to analyze, providing credentials

for authentication, providing additional parameters, executing analysis, and finally, reviewing the results. Conduct the following steps on the SQL Server system you plan on upgrading in order to perform a compatibility test using the SQL Server 2008 Upgrade Advisor:

1. Click Start, All Programs, SQL Server 2008, SQL Server 2008 Upgrade Advisor.
2. On the Welcome to SQL Server 2008 Upgrade Advisor page, select the link Launch Upgrade Advisor Analysis Wizard, and then click Next.
3. On the SQL Server Components page, provide the name of the SQL Server and then specify the components that will be analyzed. Click Next as displayed in Figure 1.6.

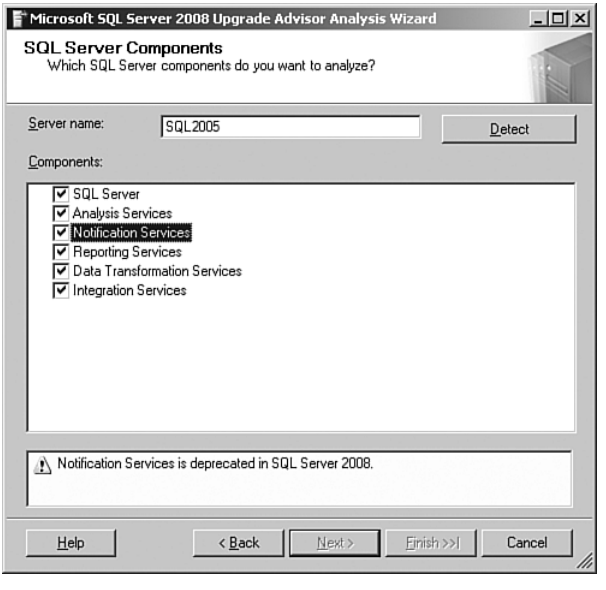

FIGURE 1.6
Specifying the SQL Server Components to analyze.

4. Provide the authentication mechanism for the SQL Server instance and then click Next.
5. On the SQL Server Parameters page, indicate what additional elements should be analyzed. Options include databases, trace files, and SQL Server batch files. Click Next to continue.

6. The Reporting Services Parameters page is an optional step. If a reporting services scan was selected, enter the name of the Reporting Services instance and then choose the authentication mechanism that will be used. Click Next.
7. The Analysis Services Parameters page is an another optional step. If an Analysis Services scan was selected, enter the name of the Analysis Services instance. Next, choose the authentication mechanism that will be used. Click Next.
8. The next optional screen focuses on DTS parameters for DTS packages. Select either the Analyze DTS Packages on Server option or the Analyze DTS Package Files option. If the second option is selected, specify the path to the DTS packages. Click Next to continue.
9. On the final page, SSIS Parameters, indicate whether you want to analyze SSIS packages on the server or files. If the second option is selected, specify the path to the SSIS packages. In addition, if the packages are password protected, enter a password. Click Next to continue.
10. Confirm the Upgrade Advisor Settings and then click Run to commence the analysis.
11. The Upgrade Advisor Progress page provides progress messages for each component being analyzed. The status message includes any of the words error, failure, or success. View the status messages in the details pane or alternatively, launch the report. Click Close as indicated in Figure 1.7.

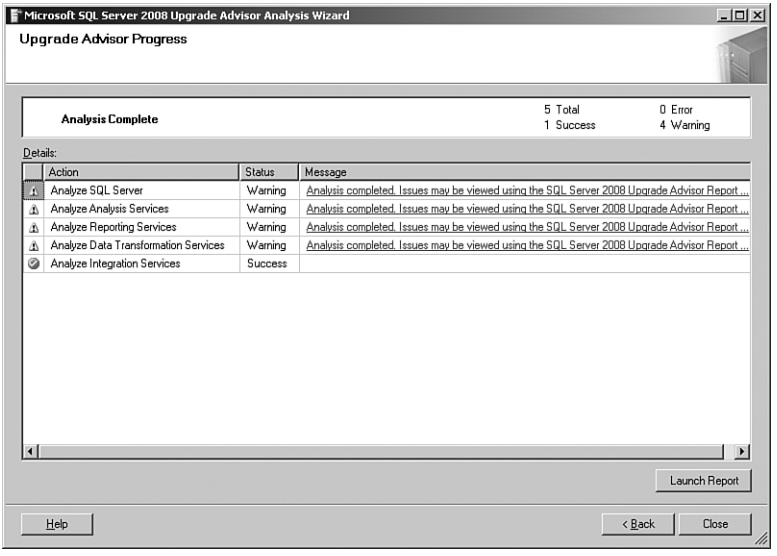

FIGURE 1.7
Reviewing the Upgrade Advisor Analysis Complete Report.

> **Note**
>
> The analysis output is written to a file; therefore, the report can be viewed from the Upgrade Advisor Progress page or at a later date. To review the report at another time, launch the Upgrade Advisor Report Viewer from the Upgrade Advisor start page.

The report can be viewed by server and then by instance or component. Moreover, the report can be filtered by All Issues, All Upgrade Issues, Pre-Upgrade Issues, All Migration issues, Resolved Issues, or Unresolved Issues. The output report also indicates when issues should be addressed. For instance, the report may indicate the issue should be addressed before the upgrade or after the upgrade. It is beneficial to review each message to ensure there are no issues when upgrading the existing SQL Server system to SQL Server 2008. When drilling through each message, it is possible to expand upon an issue and gain additional information on how to resolve the anomaly by clicking the link titled Tell Me More About This Issue and How to Fix It.

Additional Considerations Before Upgrading the Database Engine to SQL Server 2008

The following additional considerations apply before upgrading the Database Engine to SQL Server 2008:

- Even though this book does not focus on Analysis Services, if you are upgrading a 64-bit edition of SQL Server, it is imperative to upgrade Analysis Service first and then the Database Engine.
- Run the appropriate `DBCC` commands to ensure that both the system and user databases are in good health. A maintenance plan can be generated to complete these tasks.
- Make certain that all databases, specifically the system databases, are configured to autogrow. The system databases includes master, model, msdb, and tempdb.
- Ensure that you have administrative access to all user and system databases and that each database has logon information in the master system database.
- Configure the Max Worker Threads setting to a value of 0.
- Disable all startup stored procedures as the upgrade process may restart the server.
- If Replication is enabled, stop replication during the upgrade process.

- Conduct a rolling upgrade if Database Mirroring is used. First upgrade the mirrored instance, failover services, and then upgrade the principal instance (which is now the mirror). It is also recommended to remove the witness and change the operation mode to high safety during the upgrade.
- In SQL Server 2000, Log Shipping was established with a Database Maintenance Plan. Because the installation in SQL Server 2005 and SQL Server 2008 no longer uses a maintenance plan to implement Log Shipping, it is not possible to upgrade a SQL Server 2000 system running log shipping to SQL Server 2008.

Performing the SQL Server 2008 Upgrade

At this point, you have accomplished quite a few tasks. Let's review: your data is backed up, you have read the release notes, you ran the SQL Server System Configuration Checker and the SQL Server Upgrade Advisor, and you addressed the issues or warnings identified. It is now time to upgrade to SQL Server 2008.

1. Log in to the server and insert the SQL Server 2008 media. Autorun should launch the SQL Server 2008 Installation Center landing page; otherwise, click Setup.exe.
2. On the SQL Server Installation Center landing page, first select the Installation link, and then Upgrade from SQL Server 2000 or SQL Server 2005.
3. On the Setup Support Rules page, review the outcome of the System Configuration Checker. Ensure that all tests associated with the operation passed without any failures, warnings, or skipped elements. Alternatively, you can review a standard or comprehensive report by selecting the Show Details button or View Detailed Report. Click OK to continue with the installation.
4. On the Setup Support Files page, the wizard will prompt whether or not additional setup support files are required for the installation. If additional files are required, review the status of the files required and click Install.
5. The Setup Support Rules page will be displayed again and will identify any outstanding items that may hinder the installation process associated with the SQL Server cluster. Review and correct failures and warnings before commencing the installation. If failures are not displayed, click Next to start the installation.

6. On the Product Key page, enter the SQL Server Product Key and click Next.
7. On the License Agreement page, accept the Licensing Terms, and then click Next.
8. On the Select Instance page, use the drop-down menu and specify a SQL Server instance to upgrade. Click Next as displayed in Figure 1.8.

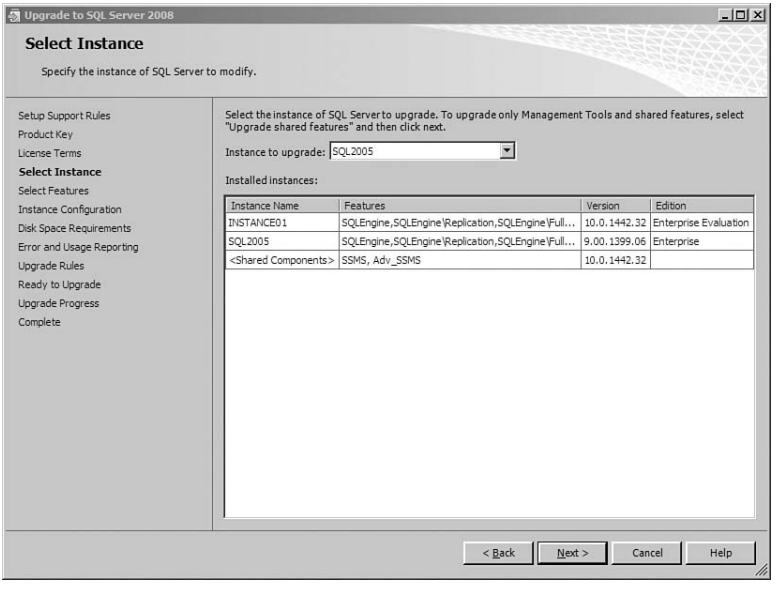

FIGURE 1.8
Specify the SQL Server instance to upgrade.

Note

The Installed Instances section displays all the instances installed on the system. In addition, to upgrade only SQL Server Management Tools, choose the option Upgrade Shared Features Only in the Instance to Upgrade drop-down list.

9. Review the features to be upgraded in the Select Features page, and then Click Next.

Note

It is not possible to modify the SQL Server features being released during an upgrade.

10. Review the name and InstanceID for the SQL Server instance being upgraded and click Next.
11. The next page is the Disk Space Requirements. Review the disk space summary for the SQL Server components and features selected to be upgraded, and then click Next.
12. On the Full-Text Upgrade page, specify an option of how the existing Full-Text catalogs will be processed after the upgrade. Click Next as displayed in Figure 1.9.

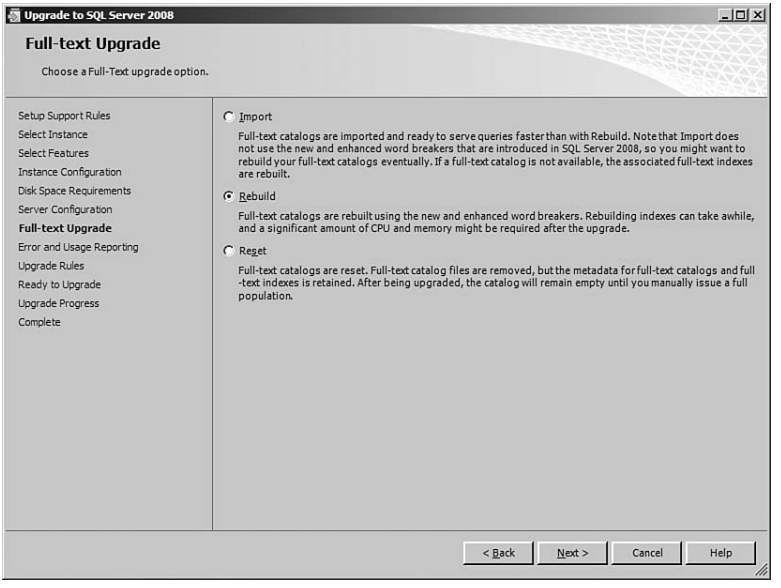

FIGURE 1.9
Specifying the Full-Text Upgrade option.

- **Import**—Full-Text catalogs are imported "as is" and are ready to serve queries. This process is much faster than rebuilding the Full-Text catalogs; however, the import does not leverage the new Full-Text features and functionality associated with SQL Server 2008.

- **Rebuild**—When this option is selected, the Full-Text catalogs are rebuilt using the new and enhanced word breakers associated with SQL Server 2008. This process is typically slower, and a significant amount of processor and memory will be required to tackle this operation.
- **Reset**—The final option is Reset. The Full-Text catalogs will be removed; therefore, after the upgrade is complete, the catalogs will remain empty until they are manually processed with a full population.

13. On the Error and Usage Reporting page, help Microsoft improve SQL Server features and services by sending error reports and feature usage to Microsoft. Specify the level of participation, and then click Next.
14. The final check will take place to ensure that the upgrade process will not be blocked. On the Upgrade Installation Rules page, review for any outstanding errors or warnings and then click Next to continue.
15. Before commencing the SQL Server 2008 upgrade, review the features to be upgraded on the Ready to Upgrade page, and then click Install.
16. When the upgrade process starts, you can monitor its progress accordingly. When the upgrade setup completes, review the success status based on each SQL Server feature. Click Next as illustrated in Figure 1.10.

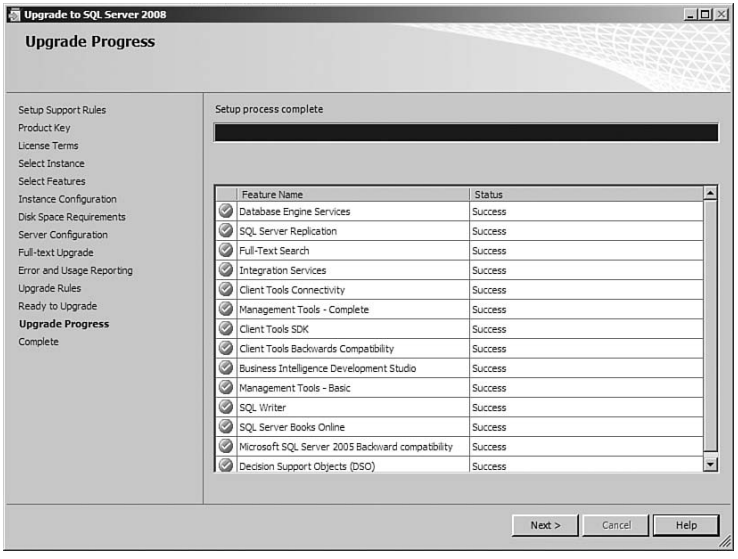

FIGURE 1.10
Reviewing the results of the SQL Server 2008 upgrade.

17. On the Complete page, review the location of the SQL Server summary upgrade log file and items in the Supplemental Information section. Click Close to finalize the upgrade.
18. To conduct post-installation tasks, review the upcoming section "Finalizing the SQL Server 2008 Installation or Upgrade."

Finalizing the SQL Server 2008 Installation or Upgrade

Once the installation or upgrade of SQL Server 2008 is complete, it is beneficial to review the following tasks in order to finalize the installation.

Reviewing SQL Server 2008 Logs

When the installation or upgrade is complete, it is best practice to review the setup log file, review the Windows application log, and review SQL Server logs for any issues or warnings. As noted earlier, the location to the SQL Server 2008 installation setup file can be found on the Complete page during the final stages of the installation or upgrade.

Downloading and Installing Updates

Even though the Windows Server 2008 system may be configured to automatically obtain server updates, it is still a best practice to check for missing SQL Server 2008 and Windows Server 2008 service packs and critical fixes. These outstanding service packs and critical fixes can be installed with Microsoft Update or a software distribution tool such as System Center 2007 Configuration Manager.

Hardening the SQL Server Installation

Another important step to finalize the SQL Server 2008 installation or upgrade is hardening the SQL Server implementation. There are a number of tasks that should be completed in order to harden the SQL Server installation. Some of these tasked include using Policy Based Management to reduce the surface attack area, enabling a Windows Server 2008 advanced firewall, and leveraging the SQL Server Configuration Manager Tool to disable unnecessary protocols and features.

Note

The SQL Server Surface Area Configuration Tool has been deprecated in SQL Server 2008. Therefore, Policy Base Management and `sp_configure` should be utilized instead, in order to harden and configure the surface area of a SQL Server installation.

Items to Consider After an Upgrade

This section describes additional items to take into consideration after an upgrade to SQL Server 2008 is complete.

Running SQL Server Management Studio for the First Time

After the upgrade is complete and you launch SQL Server Management Studio for the first time, you will be prompted to import customized user settings from SQL Server 2005 Management Studio. Click Yes or No and be aware that some SQL Server 2008 default settings might be changed after you import your customized settings.

Choosing the Database Compatibility Level After the Upgrade

When SQL Server systems are upgraded to SQL Server 2008, it is beneficial to understand how compatibility level settings affect databases. The compatibility levels include

- SQL Server 2008—Version 100
- SQL Server 2005—Version 90
- SQL Server 2000—Version 80

If you select one of these options, the database behaviors are to be compatible with that specified version of SQL Server. This setting only affects a specific database and not all databases associated with a SQL Server instance.

Note

After the upgrade, SQL Server automatically sets the compatibility level to the earlier version of SQL Server.

The settings can be changed by right-clicking a database and specifying the compatibility level on the Database Options page or by using the `ALTER DATABASE` command. The following Transact-SQL sample illustrates how to change the compatibility level.

```
Alter Database <database name>
Set Compatibility_Level =<80 ¦ 90 ¦ 100>
```

From a best-practice perspective, it is recommended to change the database to single-user mode before changing the database compatibility settings. This prevents inconsistent results if active queries are executed.

Additional Post-Upgrade Tasks

- Update Statistics on all users and system databases.
- Execute `DBCC_UPDATEUSAGE` on all databases to ensure that all databases have the correct row and page counts.
- With SQL Server 2008, queries on partitioned tables and indexes are processed differently. Therefore, it is recommended to remove the `USE PLAN` hint from the query.

Managing SQL Server 2008 Installations

The following sections explain how to manage SQL Server 2008 installations.

Employing Additional SQL Server 2008 Instances

As mentioned earlier, many organizations decide on scaling up their SQL Server infrastructure by creating consolidated SQL Server systems with multiple-instance installations. To achieve the goal of installing additional instances on an existing system, a DBA must relaunch the SQL Server 2008 installation utility, and then select the option New SQL Server Stand-alone Installation or Add Features to an Existing Installation.

When the new SQL Server installation wizard begins, follow the steps in the earlier section "Installing a Clean Version of SQL Server 2008"; however, on the Installation Type page, select the option Perform a New Installation of SQL Server 2008, as displayed in Figure 1.11. Then on the Feature Selection page, select the desired features to be installed for the new instance. Finally, on the Instance Configuration page, provide the instance with a unique name and proceed with the installation.

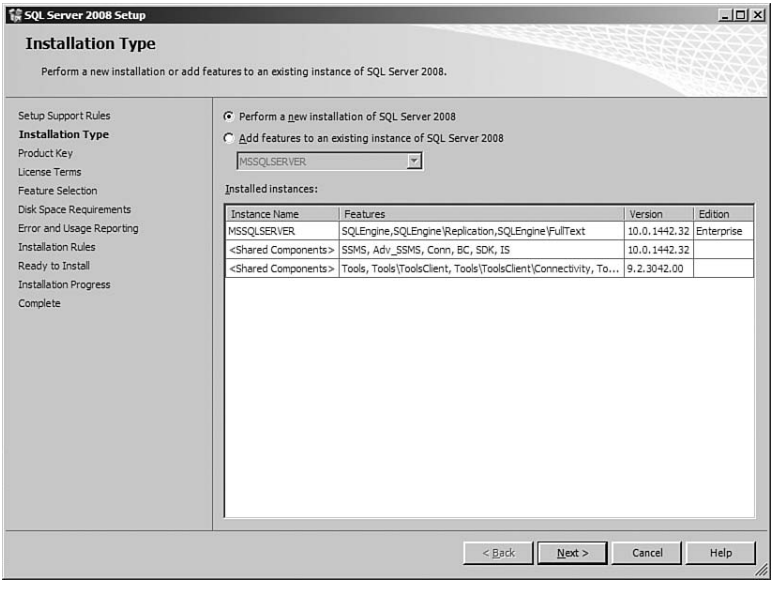

FIGURE 1.11
Adding additional SQL Server instances to an existing installation.

Adding Features to an Existing SQL Server 2008 Installation

The process for adding and removing SQL Server features to an existing Installation is similar to the steps involved when adding additional SQL Server instances. The DBA must select New SQL Server Stand-alone Installation or Add Features to an Existing Installation from the SQL Server 2008 Installation Center's Installation page. However, on the Installation Type screen, the option Add Features to an Existing Instance of SQL Server 2008 must be selected. Then on the Feature Selection page, select the features to be added and continue through the wizard.

> **Note**
>
> It is not possible to add features when upgrading to SQL Server 2008; therefore, this strategy should be used for adding additional features after the SQL Server upgrade is complete.

Changing SQL Server 2008 Editions

Another feature included with SQL Server 2008 is the potential to conduct an Edition upgrade after SQL Server 2008 has been installed. For example, if an organization is running the Standard Edition and decides that they want to leverage the features and functionality associated with the Enterprise Edition, they simply conduct an edition upgrade instead of formatting and reinstalling from scratch. Another advantageous scenario includes moving from SQL Server 2005 Standard to SQL Server 2008 Enterprise Edition. This objective would be achieved by first upgrading the SQL Server system from SQL Server 2005 Standard to SQL Server 2008 Standard, and then running the Edition Upgrade to upgrade the installation Enterprise Edition of SQL Server 2008.

To conduct an Edition Upgrade on SQL Server 2008, the Edition Upgrade must be selected from the Maintenance page on the SQL Server 2008 Installation Center landing screen.

Summary

The SQL Server 2008 installation process and deployment tools bear similarities to those found in previous versions of SQL Server. However, feature and performance enhancements associated with the new SQL Server 2008 Installation Center tool have improved the installation experience—whether you are installing a single SQL Server implementation from scratch or upgrading an existing system to SQL Server 2008.

Best Practices

The following are best practices from this chapter:

- Verify that your hardware, devices, and drivers are supported by SQL Server 2008.
- Stick to using the recommended or optimal hardware and software requirements when installing or upgrading to SQL Server 2008.
- Leverage the planning tools and documentation associated with the SQL Server Installation Center when installing or upgrading to SQL Server 2008.
- Run the System Configuration Checker tool as a prerequisite task when either installing or upgrading to SQL Server 2008.
- Install and run the Upgrade Advisor to identify any upgrade anomalies when upgrading a system to SQL Server 2008.

- When performing an upgrade, make sure you document your SQL Server system and database configuration information and perform a backup of any SQL Server data and objects that you want to keep.
- Leverage Windows Server 2008 as the operating system when running SQL Server 2008.
- Finalize a SQL Server implementation by hardening the system based on the best practices listed in Chapter 8.
- Because the SAC tool has been deprecated, utilize Policy Based Management to configure the surface area of one or many SQL Server systems.
- The Windows Server 2008 Advanced Firewall is enabled by default; therefore, review Chapter 8 to understand how to configure the firewall for SQL Server access.
- Review Books Online if you need to upgrade other SQL Server 2008 features above and beyond the Database Engine.
- Data, log, and tempdb directories should be on separate physical disks or Logical Unit Numbers (LUNs) for performance whenever possible.

Index

A

B

C

D

E

F

G

H

I

M

N

O

P

Q

R

S

T

U

V

W

X