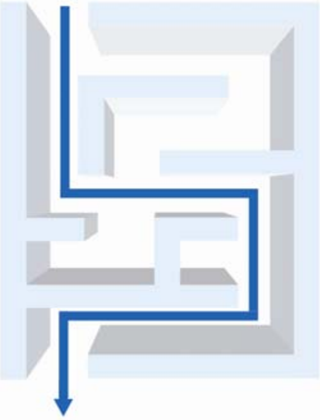
DEPLOYMENT



CONSOLIDATING SQL SERVER™ 2000 AND SQL SERVER™ 2005 DATABASES TO SQL SERVER™ 2008 ON WINDOWS SERVER™ 2008 ENTERPRISE ON DELL™ SERVERS

GUIDE

A Principled Technologies report commissioned by Dell

Table of contents

Introduction

Consolidation overview

Dell PowerEdge 2950

This Consolidation Guide is the first in a series of documents that will provide concepts, procedures, and hard performance data that will help you successfully consolidate your Microsoft® SQL Server® 2000 and SQL Server 2005 databases from multiple machines onto a single SQL Server 2008 server system. In this Guide, we use the Dell™ PowerEdge™ 2950 as a target server, but upcoming guides will feature other Dell PowerEdge servers.

PT has performed hands-on testing and research and drawn on real-world experiences to document best practices and help systems administrators and database administrators simplify operations and take advantage of new features in SQL Server 2008.

While upcoming guides will give hard performance data to assist you in your sizing efforts, this document covers the key overall concepts and procedures. It reviews in detail the approach that Principled Technologies® (PT) and Dell recommend for a tested and validated consolidation of SQL Server 2000 and SQL Server 2005 databases to SQL Server 2008 using the Dell PowerEdge 2950 running Microsoft Windows Server® 2008 Enterprise.

What is consolidation?

Generally speaking, *consolidation* refers to the process of combining multiple things to make a single, more effective unit. In an IT context, you can consolidate several types of things, including the following:

Physical servers. In this Guide, consolidation typically means consolidating servers. After a successful consolidation, all applications should run on fewer servers than before, and those applications should run at least as well as they did previously. The other types of consolidation we mention below may happen as a consequence of consolidating physical servers, but they are not our primary focus. Upcoming guides will give hard performance data and guidelines for how many servers of a particular type you can expect to consolidate to a single given PowerEdge server.

SQL Server instances. As you consolidate servers, you will have the choice of installing single or multiple instances of SQL Server. While we share below some guidelines for making this decision, your specific requirements must guide your choice.

Storage. As you follow the procedures in this Guide, you will consolidate storage to some degree by moving the data from a number of servers to a large, locally attached, disk storage subsystem in a new server. Larger-scale storage consolidations, such as implementing a storage area network (SAN), are outside the scope of this guide.

Locations. As you consolidate servers, you may reduce the number of locations that house servers. Planning for consolidating geographic locations is also beyond the scope of this guide.

As we stated above, in this Guide we discuss how you can consolidate multiple SQL Server databases onto smaller numbers of physical servers. You can do this in several ways:

- combine databases into a single instance of SQL Server
- run multiple instances of SQL Server on a single physical server
- use virtualization to give the appearance of several physical servers

Though each of these approaches has its merits, in this guide we cover only the first two. We explain how to consolidate multiple databases to a single instance of SQL Server on your PowerEdge 2950, and we explain how to create additional instances of SQL Server on that machine. Virtualization, though a powerful tool for consolidation, is outside the scope of this guide.

Why consolidate?

Done well, consolidation yields a more economical environment with more consistent management, better use of hardware, and all applications performing at least as well as they did previously. An effective consolidation effort improves reliability and security while bringing great savings to the organization. Some of the ways consolidation can stretch your technology dollar include the following:

Hardware savings

Buying, powering, and supporting fewer servers brings obvious savings to an organization. Less obvious are the other significant cost savings that fewer servers mean over time, such as fewer hardware replacement purchases as these consolidated servers age out. In other words, your future purchase growth rate may decrease. Other potential hardware cost savings areas include racks and network switches: as the number of servers decreases, these costs decrease as well.

Software licensing savings

Consolidation often saves an organization significant money in software licensing. Licensing varies by product, but to see the possibilities, consider this example.

NOTE: As we write this Guide, Microsoft has not released SQL Server 2008 licensing information, so this example uses SQL Server 2005 licensing prices.

Microsoft currently offers three SQL Server 2005 licensing models: Server plus User Client Access License (CAL), Server plus Device CAL, and per-processor licensing. The first two models require a server fee and a license per user/device. The third, which Microsoft intends for servers with large numbers of users, charges a flat rate per processor socket on your SQL Server hardware. (Details on these models are available at http://www.microsoft.com/sql/howtobuy/licensing.mspx.)

Consolidating servers increases the number of clients each server handles, which means you can change your licensing strategy and achieve significant savings. Figures 1 and 2 present an example of cost savings using the per-processor licensing model. They assume five Enterprise Edition SQL Server instances, each running on a dedicated two-socket server, and they use list prices.

	Non-consolidated environment	Consolidated environment
Description	5 two-socket dual-core servers, each running SQL Server enterprise	1 four-socket quad-core server, running SQL Server Enterprise
Cost	2 sockets x $25,000 per-processor retail enterprise license x 5 servers = **$250,000** in SQL Server licensing. With 20 cores in the server, that's $12,500 per core.	4 sockets x $25,000 per-processor retail enterprise license x 1 server = **$100,000** in SQL Server licensing. With 16 cores in the server, that's $6,250 per core.

Figure 1. Comparing SQL Server list licensing costs: five two-socket servers versus one four-socket server.

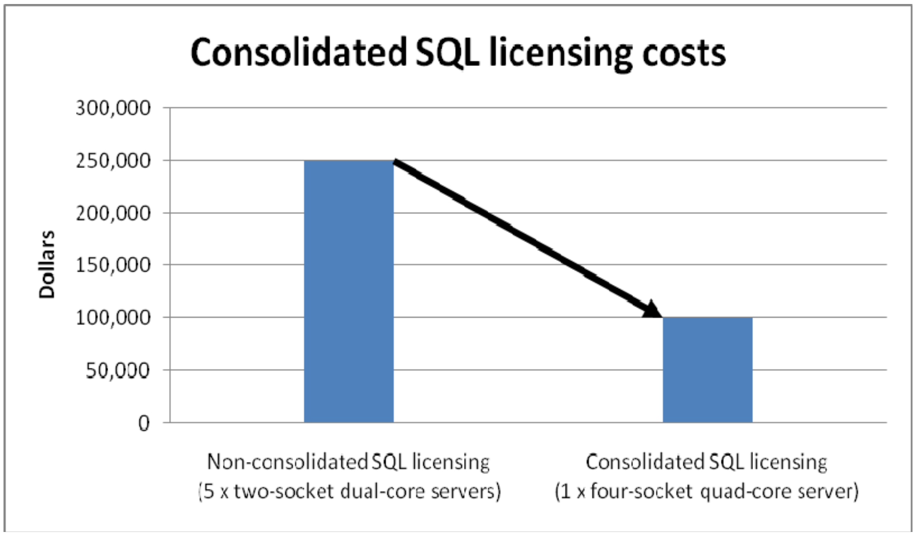


Figure 2. Example SQL Server licensing savings when consolidating from five two-socket servers to one four-socket server.

NOTE: For licensing purposes, Microsoft counts one socket as a processor, so in this example a quad-core processor counts as only one processor.

Maintenance and staff savings

A consolidated infrastructure offers many opportunities for maintenance, support, and staffing cost savings. Less hardware and associated equipment means fewer servers that require security patches and monitoring.

Reduced cost of tools. Many monitoring tools have a per-installation cost. For example, Microsoft System Center Operations Manager 2007 requires a server license for each management server and an operations management license for each managed device (see http://www.microsoft.com/systemcenter/opsmgr/howtobuy/default.mspx). Other third-party products to monitor SQL Server follow similar licensing strategies and require you to license them for each server you want to monitor. Reducing the number of servers you have to monitor reduces the cost of these tools as well.

Reduced staffing requirements. Regardless of your server-to-full-time-employee (FTE) administrator ratio, fewer servers means fewer system administrators. Also, consolidation efforts often improve standardization across servers, simplifying administration for the Windows® system administrator, network engineer, and SQL Server database administrators (DBA). Consolidation can also provide greater opportunities for automation, further reducing the FTE requirements. Because the growth in the number of servers can slow when you consolidate, these savings can compound over time.

Reduced support costs. The cost of a given level of support is usually proportional to the size of the installation. By reducing the number of servers, the costs of your support agreements go down as well.

Power and cooling savings

Consolidating servers saves power in several ways:

Fewer servers. Obviously, fewer servers consume less power.
More efficient servers. Modern servers are typically more efficient than those of a few years ago, so you are likely to consume less power per server.
Less air conditioning. Fewer, more efficient servers produce less heat. Moreover, you can house them in a smaller space, thus reducing air conditioning costs even further.

Rent savings

Consolidation reduces the floor space IT requires. By reducing the number of servers, you reduce the space required to hold the racks and the associated space necessary for servicing and airflow. You will require less cooling and so will save on the space the air conditioners would require. Finally, consolidation may allow you to reclaim space, make more efficient use of the space you have, and thus delay expansion.

Regulatory compliance savings

With fewer physical devices storing data and more uniform management, a consolidated environment can make the process of complying with regulatory requirements such as Sarbanes-Oxley and the Health Insurance Portability and Accountability Act (HIPAA) less expensive, easier, and more secure.

Sizing and baseline performance

Sizing is one key to a successful consolidation. The process of sizing involves gathering different performance baselines so you have at least an approximate set of requirements the new consolidated servers must meet. You do this by looking at the performance characteristics of existing hardware during normal business operations, and then applying growth and scalability estimates. The performance baselines you gather are useful when you configure such components of your new consolidated SQL Server environment as SQL Server 2008’s Resource Governor.

Among the characteristics to examine on each server are the following:

- Processor utilization
- Memory requirements of the OS and applications

- Disk layout
- Database size
- Expected database growth
- Maximum concurrent users
- Types and rates of transactions against the databases

The server you choose for consolidation must do more than match the combined capacity of all the other current servers today. It must have enough excess capacity to still perform well at the end of its expected life span. Fortunately, given the improvements we note below, finding such servers is entirely possible.

Another reason for consolidation is one that plays a role in sizing: the typical enterprise is not using its servers at anywhere near capacity. As Ben Pring, Research Vice President for Gartner, noted at the Gartner Symposium ITxpo 2007, "Server utilization was also low—at an average 18 percent in large organizations."

In recent years, server capabilities have greatly increased. Here are some of the improvements that make it possible to consolidate multiple servers into a single system:

More processors. Modern quad-core processors double or even quadruple the number of processing cores available without increasing the physical size of the server.

More processing power per core. While clock speeds have changed little, processors have undergone a generational shift. Industry-standard benchmarks, such as SPECint, show today's processors are much faster than their counterparts of three to five years ago.

64-bit address space. The advantages of the vastly greater address space on a 64-bit environment are widely documented. Five years ago, many processors did not support 64-bit addressing. Windows Server 2003 did not support 64-bit addressing on x64 processors until 2005.

Faster memory. The memory in a modern system is two to three times as fast as that in a three- to five-year-old system.

Faster storage. Disk transfer rates have improved substantially.

Cheaper storage and greater capacity. Disk capacity has greatly increased, while the cost per GB has decreased. Moreover, external storage arrays radically increase the amount of storage available to a server without incurring the cost of a SAN.

We further discuss sizing at a conceptual level in the Getting ready section. The upcoming consolidation guides we discussed earlier will provide detailed sizing examples.

New features of SQL Server 2008

With the launch of SQL Server 2008, Microsoft introduced a wide variety of new features while enhancing the existing capabilities of SQL Server 2005. There are far too many features to discuss in this Guide, but in this section we highlight some of the more significant new SQL 2008 benefits that relate to consolidation.

Resource Governor

SQL Server 2008's Resource Governor allows you to control the amount of CPU and memory available to a given workload group. Previously, a single badly behaved application or query could negatively affect other users. Resource Governor lets you define groups and limit the maximum resources available to those groups. This technique makes it very difficult for a single user, group of users, application, or query to monopolize the server's hardware resources. Resource Governor also lets you grant minimum resource levels to high-priority users, ensuring that database resources accurately reflect your business needs and priorities. This ability to control resource allocation at a granular level protects lower priority processes and applications from resource starvation, while still guaranteeing high-priority processes the resources they need. We provide examples of working with Resource Governor in the Setting up Resource Governor section.

Declarative Management Framework

SQL Server's Declarative Management Framework helps you implement policy-based management across the enterprise. Policies for consolidated SQL servers are generally more complicated than those for dedicated servers. Naturally, as you increase the number of databases, users, applications, and developers in a consolidated environment, your configuration needs more structured policy management than when each database server had a single purpose. Declarative Management allows the database administrator to set strict policy guidelines and also manage actions when users violate those policies. Declarative Management lets you manage everything about your SQL Server instances, including backup devices, indexes, tables, procedures and triggers, database options, server options, and more.

Declarative Management is a powerful and flexible framework for defining the policies that govern everything concerning your server and your databases. You can store, export, and import policies, making it much easier to use a consistent set of policies across your business.

Auditing and tracking data change

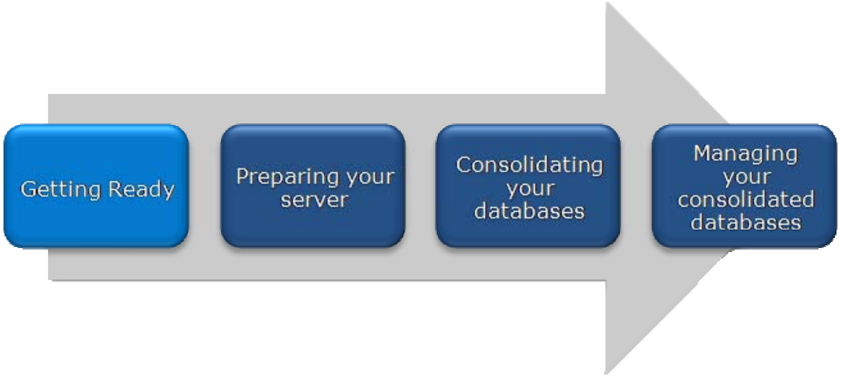
Auditing in SQL Server 2008 Enterprise Edition is more seamless than in previous versions. You can create customized audits for the various properties and settings throughout your SQL Server instances and at the database level. You can configure SQL Server auditing to stream your audit output to the event log or a filesystem file.

Two important data change features are also present in SQL Server 2008: Change Tracking and Change Data Capture. These features allow data administrators to monitor and capture changes in the physical data: who changed the data, when they changed it, and what exactly they changed.

Built-in data compression

Compressing data lets you trade speed for space: accessing compressed data is slower and requires more CPU resources than accessing uncompressed data. In certain scenarios, compressing data is a good choice, because compressing data can save enough space to let you comfortably consolidate that data with another server.

Getting ready


Getting Ready
Preparing your server
Consolidating your databases
Managing your consolidated databases

Evaluating your databases for consolidation

You must be aware of many specific details related to each database you target for consolidation, including the maintenance window in which you move the database, the users the move affects, and the configuration tasks necessary to assimilate the database into your consolidated environment. A more comprehensive example survey appears in Appendix A. Information to gather before consolidation may include the following:

- Server OS version and patch level
- Hardware specifics
- SQL Server version and patch level
- Number of users in this database and what type of logins (Windows or SQL) map to these users
- Current backup strategy and schedule for this database
- Clustering information, if any
- Replication or mirroring details, if any
- Detailed information regarding permissions and roles
- SQL Agent jobs that reference this database

Gather baseline performance data

During your research phase, you should use Performance Monitor, SQL Server Profiler, and other tools to gather data on the typical query load and performance statistics for the databases you're considering consolidating. This practice serves two purposes. First, it provides a prime opportunity to find potential problems before you consolidate. Second, you can use the information you gather to map out your resource pool needs and workload groups to use with SQL Server 2008's Resource Governor feature.

"Move up" when possible

When possible, you should attempt to upgrade features and hardware so that your new consolidated configuration is as efficient, powerful, consistent, and current as possible. Some examples include the following:

Upgrade your hardware. To ensure that performance remains optimal for users, your consolidated server should utilize the latest and most powerful processors, memory, and disks that your budget permits. You should also optimize and upgrade other auxiliary equipment such as storage arrays.
Move your databases to a 64-bit environment. Doing so lets you take advantage of the larger virtual address space the 64-bit environment offers.
Use the latest operating system version and SQL Server version. Doing so lets you leverage important new features that help with managing a consolidated environment. In this Guide, we use Windows Server 2008 and SQL Server 2008.

Poor candidates for consolidation

Not every database is a good candidate for consolidation. Here we present examples of databases that might do best on dedicated servers.

Databases with critical performance requirements. If your database has stringent performance requirements, it may be best to leave it on a dedicated server. On a dedicated server, it is much easier to find any bottlenecks that might occur. It is also much simpler to address any problems when you need not worry about disrupting other work on the same server. If you do decide to proceed with consolidation, try to avoid consolidating two performance-critical databases on the same server, because it may not be possible to achieve acceptable performance for both of them.

Databases with critical uptime requirements. Avoid consolidating multiple highly critical databases on a single server. To do so greatly magnifies the effect of a single server failure. While high-availability measures such as mirroring or clustering can help mitigate this risk, it's often better to not have too many critical resources vulnerable to a single point of failure.

Databases with critical security requirements. You may need to isolate databases with highly sensitive data. You may be able to handle some critical requirements by putting such a database in a dedicated SQL Server instance, but you may decide that some databases are sufficiently sensitive that you need to keep them on dedicated servers.

Databases with legacy requirements. Some databases may have requirements you cannot fulfill in the new environment, such as applications that depend on highly specialized file layouts or on services that have no equivalent in SQL Server 2008. Examples include databases involved in carefully configured multi-tier applications, those you plan to age out in the near future, or those with third-party front-end applications that have special requirements.

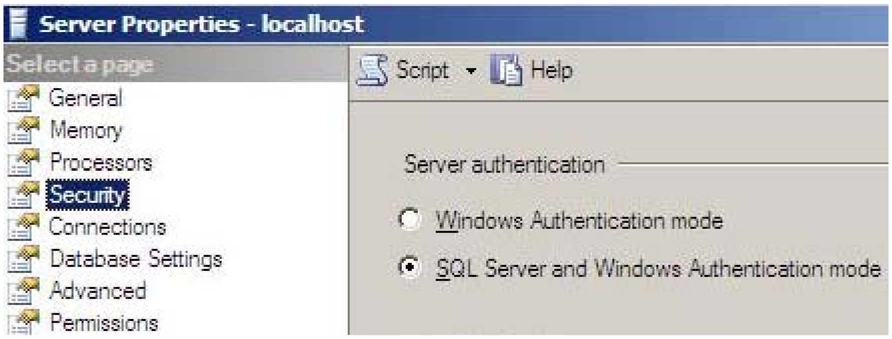
Of course, you must also be sensitive to circumstances unique to your organization. Contractual obligations, for example, might require you to keep a given database on an isolated server.

Finally, avoid mixing contrasting types of work. For example, the usage patterns and schema designs for OLAP and OLTP differ significantly. Each of these environments employs different optimization techniques, and those techniques may not mix well. If you consolidate your OLAP relational data warehouse alongside your OLTP database, you may find that OLTP performance suffers when bulk data loading or report generation occurs against your data warehouse databases.

When to use multiple instances

Like SQL Server 2000 and SQL Server 2005, SQL Server 2008 lets you install multiple named instances in addition to the default instance. At a software and security level, each instance of SQL Server is independent of any other instance on the system. An instance has its own system databases and security context, and it can listen on a specific TCP port. Although running multiple SQL Server instances generates some overhead, in certain situations doing so can benefit your organization. Here are some examples:

Organizational, policy, or business separation. Organizations often need to completely separate the actions, queries, and application usage of certain user or development groups from others. This separation could be for security reasons, backup or maintenance time windows, or other management reasons.
Security context. Each SQL Server instance has its own security context. Logins and users you create and manage on one instance are completely independent of other instances, even when the instances are running on the same physical hardware. An example of security needs justifying two instances would be having both legacy applications that still demand the use of SQL Server authentication and new applications that leverage only Active Directory® domain authentication. You could run one instance for mixed-mode applications in SQL Server and Windows Authentication Mode and configure your other instance to use the more secure Windows Authentication Mode (see Figure 3).



Server authentication

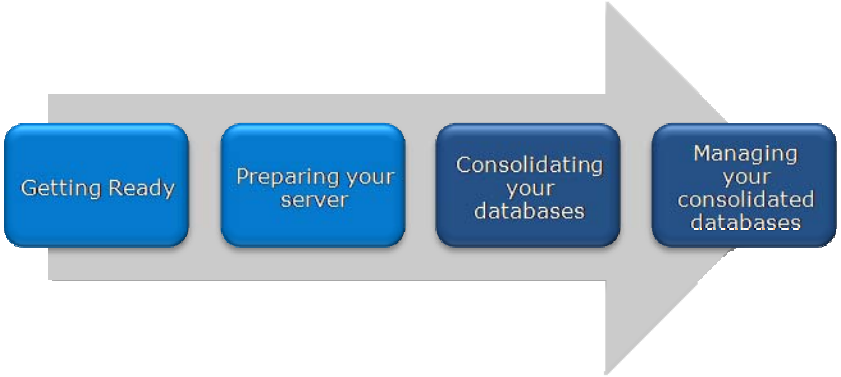
○ Windows Authentication mode

● SQL Server and Windows Authentication mode

Figure 3. Setting the authentication mode in the Server Properties window in SQL Server Management Studio.

Manageability. Depending on your enterprise, you may find that manageability is easier in certain circumstances when running multiple instances on the same hardware. For example, when you're using the new Declarative Management Framework feature in SQL Server 2008, you may apply certain policy sets on some instances and other policy sets on other instances.
Testing and development. If your IT department needs multiple independent environments on which to develop and test systems, and hardware segmentation is either not necessary or not possible, multiple instances may be a good solution.

Preparing your server


Getting Ready
Preparing your server
Consolidating your databases
Managing your consolidated databases

Defining our environment

We used a Windows domain containing a SQL Server 2000 database server, a SQL Server 2005 database server, a new SQL Server 2008 database server, and an Active Directory server.

To be specific, our example consolidation database server was a Dell PowerEdge 2950 running Windows Server 2008 Enterprise Edition SP1 x64 and SQL Server 2008 Enterprise Edition x64 February CTP. We connected a Dell PowerVault^{TM} MD1000 direct-attached storage array to the PowerEdge 2950. Our first legacy server was a Dell PowerEdge 2650 running Windows Server 2003 R2 Enterprise Edition SP2 x86 and SQL Server 2000 Enterprise Edition SP4. Our second legacy server was a Dell PowerEdge 6650 running Windows Server 2003 R2 Enterprise Edition SP2 x86 and SQL Server 2005 SP2 x86. The PowerEdge 2650 and PowerEdge 6650 used only internal storage. Our Active Directory server was a Dell PowerEdge 1950 running Windows Server 2003 R2 Enterprise Edition SP2. We connected all components via a gigabit Ethernet switch. Figure 4 illustrates our setup.

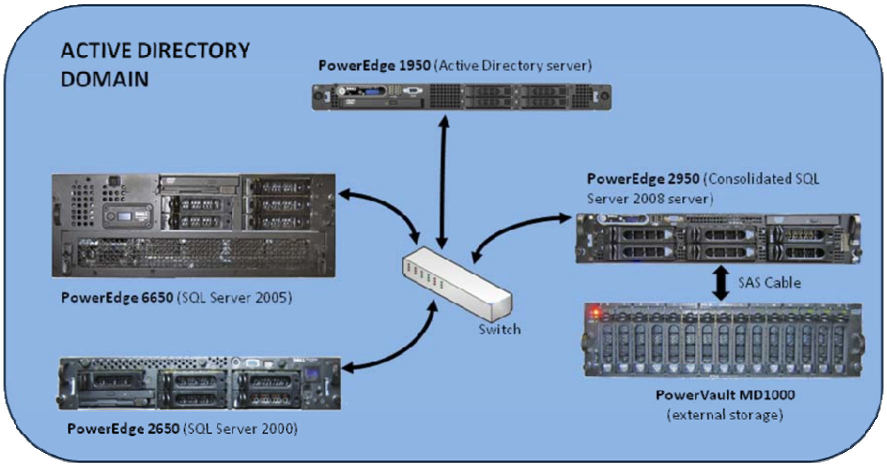

ACTIVE DIRECTORY DOMAIN
PowerEdge 1950 (Active Directory server)
PowerEdge 2950 (Consolidated SQL Server 2008 server)
SAS Cable
PowerEdge 6650 (SQL Server 2005)
Switch
PowerVault MD1000 (external storage)
PowerEdge 2650 (SQL Server 2000)

Figure 4. The setup we used in our hands-on testing and research for this Guide.

Figures 5 and 6 present the hardware and software we used to simulate an Active Directory domain and associated SQL Server 2000, SQL Server 2005, and SQL Server 2008 database servers.

Server	Processor	Memory	Disk drives
Dell 1950 (Active Directory server)	2 x Quad-Core E5440 Intel® Xeon® (2.83 GHz)	16 GB	2 x SAS, 15K RPM, 146 GB
Dell 2650 (SQL Server 2000 database server)	2 x single-core Intel Xeon (2.8 GHz)	8 GB	3 x Ultra160 SCSI, 10K RPM, 36 GB
Dell 6650 (SQL Server 2005 database server)	4 x single-core Intel Xeon (2.00 GHz)	6 GB	3 x Ultra320 SCSI, 10K RPM, 73 GB
Dell 2950 (consolidation database server)	2 x Quad-Core E5440 Intel Xeon (2.83 GHz)	32 GB	6 x SAS, 15K RPM, 73 GB

Figure 5. Servers we used in our hands-on testing and research for this Guide.

Server	Server operating system	SQL server version
Dell 1950 (Active Directory server)	Windows Server 2003 Enterprise Edition R2 SP2 x86	N/A (Active Directory Server)
Dell 2650 (SQL Server 2000 database server)	Windows Server 2003 Enterprise Edition R2 SP2 x86	SQL Server 2000 Enterprise Edition SP4
Dell 6650 (SQL Server 2005 database server)	Windows Server 2003 Enterprise Edition R2 SP2 x86	SQL Server 2005 Enterprise Edition SP2 x86
Dell 2950 (consolidation database server)	Windows Server 2008 Enterprise Edition SP1 x64	SQL Server 2008 Enterprise Edition x64 (February CTP)

Figure 6. Software we used in our hands-on testing and research for this Guide.

We used the following storage and networking devices:

Dell PowerVault MD1000—15 x SAS, 15K RPM, 146 GB disk drives
Gigabit switch—16-port gigabit Ethernet

BEST PRACTICE: Use the latest tested and validated software, firmware, and driver versions for NICs, storage arrays, and other components. You can find these software components at http://support.dell.com/support/downloads/index.aspx?c=us&l=en&s=gen.

Configuring the RAID drives

Overview

Following Microsoft's recommended best practices, we planned to separate the SQL Server transaction logs, the tempdb system database, and the user database files in our drive configuration.

NOTE: Plan on at least 10 minutes for drive configuration on the server.

Tempdb database

SQL Server 2008 creates the tempdb system database, which all databases on the server share. SQL Server uses the tempdb database for such purposes as temporary tables, table variables, work files for hash operations, sorting, building aggregates for GROUP BY or ORDER BY clauses, index builds, Service Broker operations, INSERTED and DELETED tables that triggers use, and cursor processing.

Transaction log files

Each database has one or more transaction log files. A transaction log file stores the details for all transactions in a SQL Server database. This information is critical for restoring database consistency in the event of a failure.

User database files

These data files contain the data and objects for each user-defined database. Their contents include database objects, such as tables, indexes, and stored procedures, that you can define.

BEST PRACTICE: When possible, separate the tempdb, transaction log files, and user database files onto their own disks on separate disk groups. Doing so can yield better I/O performance by ensuring that these files do not share the same physical disks.

If your system uses the full-text search features of SQL Server 2008, create full-text catalogs on their own physical drive(s).

When possible, group files with similar I/O characteristics. For example, consolidate all logs. Because heterogeneous workloads can have very different and even competing I/O characteristics, combining heterogeneous workloads can hurt overall performance.

We cabled the server’s external RAID controller to the IN port of Enclosure Management Module (EMM) number 1 on the PowerVault MD1000. We then toggled the front switch upward on the MD1000 to indicate that we were running in unified mode and turned on the enclosure.

Below we illustrate the exact drive layout we used (see Figure 7) and briefly describe each disk group. For detailed instructions on how we configured the RAID disk groups, see Appendix B.

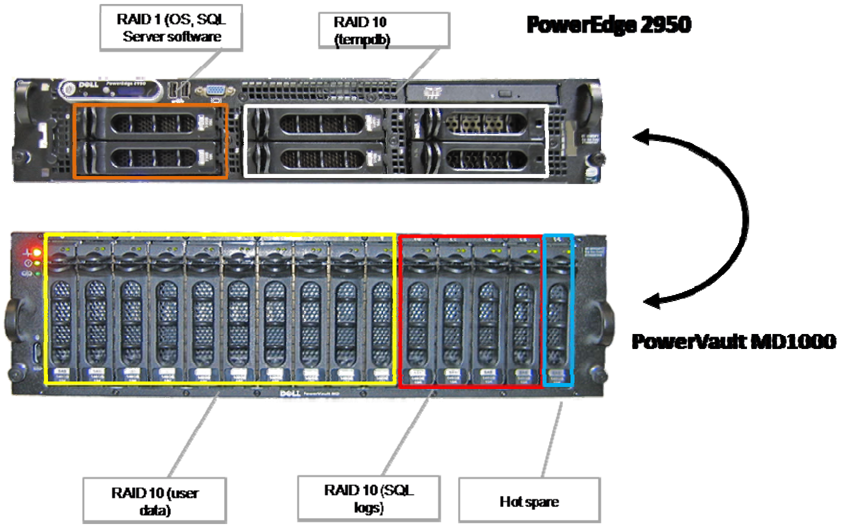


Figure 7. The drive configuration we used in the consolidated server in our hands-on testing and research for this Guide.

Internal disk drives

Our PowerEdge 2950 Server contained six drives. We configured the first two internal server drives using RAID 1 for the operating system, SQL Server 2008 software, and SQL Server system databases, with the exception of tempdb. We allocated the next four internal drives to the SQL Server tempdb files, and configured that disk group as RAID 10.

External disk drives

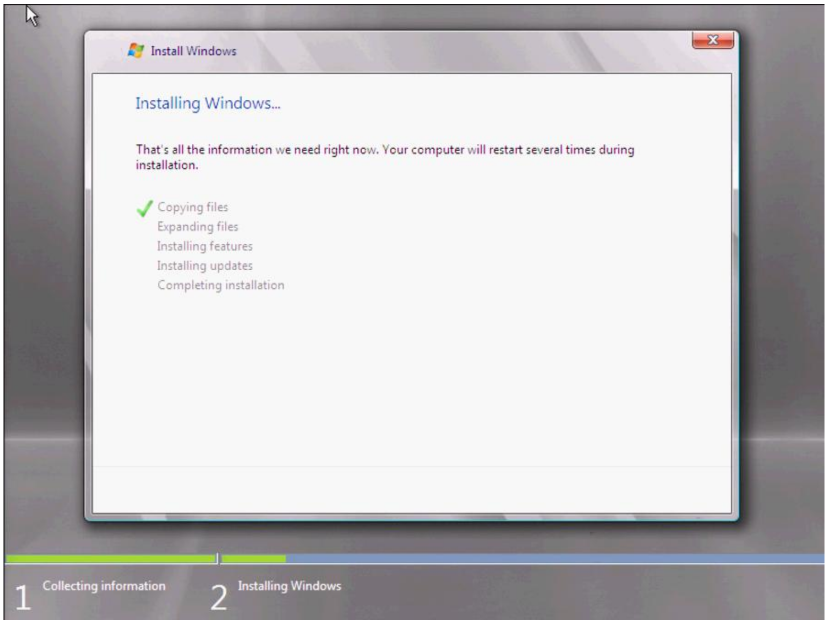
The PowerVault MD1000 contained 15 drives. We allocated 10 drives to a RAID 10 disk group that held the SQL Server user-defined databases. Then we created another RAID 10 disk group, consisting of four drives, to hold SQL Server transaction log files. We left one drive unallocated and assigned it as the hot spare.

Installing Windows Server 2008 Enterprise Edition

This section provides an overview of the operating system installation process on the Dell PowerEdge 2950. We include approximate wait times for each step. (Appendix C provides complete, detailed installation instructions.)

NOTE: Plan on at least 60 minutes for installing Windows Server 2008 Enterprise Edition on the Dell PowerEdge 2950 server. Each step below takes at least 1 minute. We provide the approximate amount of time each step takes in parentheses. These times exclude data entry time. The time to install updates—3 minutes in our setup—will increase over time as Microsoft releases additional OS updates.

1. Insert the Microsoft Windows Server 2008 Enterprise Edition DVD into the DVD drive and reboot the system. If a message appears telling you to press any key to boot from CD or DVD, do so. During the boot, a message that Windows is loading files appears, followed by a Windows booting loading bar. (3 minutes)
2. Set the basic location information, enter your product key, select full install, and enter the information the installation software requires. Start the installation (see Figure 8). (20 minutes, during which Windows resets the server twice)



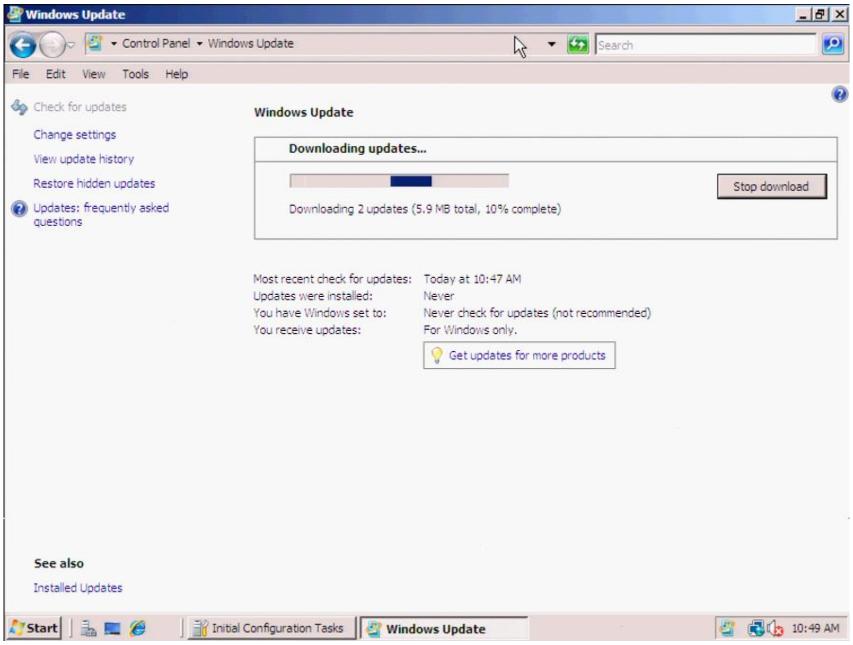
Installing Windows...

That's all the information we need right now. Your computer will restart several times during installation.

✓ Copying files
Expanding files
Installing features
Installing updates
Completing installation

Figure 8. Windows Server 2008 installation progress.

3. Set your password; enter configuration information, including the IP addresses and domain information; and reboot the system. (3 minutes)
4. Download and install updates (see Figure 9). (4 minutes; download times vary)



Downloading updates...

Downloading 2 updates (5.9 MB total, 10% complete)

Most recent check for updates:	Today at 10:47 AM
Updates were installed:	Never
You have Windows set to:	Never check for updates (not recommended)
You receive updates:	For Windows only.

Figure 9. Installing Windows Server 2008 updates.

5. Reboot after the update installation completes. (3 minutes)

BEST PRACTICES: Configure all database servers with static IP addresses. Doing so assures that SQL Server 2008 resources remain available even in the event of a DHCP server failure. It also increases the stability of your networking and DNS environments.

Use canonical names (CNAME) records in your DNS infrastructure to assign aliases to your database servers. CNAME records allow connections to use aliases for IP addresses. By using these aliases, you avoid the need to change the applications if the IP addresses change. Using CNAME records thus lets you isolate your users and applications from changes in the underlying server infrastructure. Such isolation can make deployments and migrations simpler and more robust.

Configuring the operating system drives

The next step is to configure the remaining OS drives in your server using the Disk Management utility that is part of the Windows Server 2008 Enterprise Edition Administrative Tools. Format as NTFS all SQL Server volumes, including those you use to hold your data, logs, and tempdb system database (see Figure 10).

Virtual disk	Drive location	Raid level	No. of drives	Contents
1	PowerEdge 2950	1	2	OS, SQL Server software
2	PowerEdge 2950	10	4	Tempdb files
3	PowerVault MD1000	10	10	SQL Server user data files
4	PowerVault MD1000	10	4	SQL Server log files

Figure 10. Drive configuration and contents.

Microsoft recommends the allocation unit size for these volumes be 64 KB. Should you choose to use values other than 64K, avoid values less than 8 KB, as they can increase the risk of torn pages (i.e., pages whose contents split across disk allocation units).

If you are using an allocation unit size of 64 KB, Windows Server 2008 Enterprise Edition does not offer the option to compress drives. Even if you are using another size, do not use compressed drives.

In some cases, it may be beneficial to create multiple files and filegroups for your user databases. Doing can simplify management and, in special cases, can benefit performance. In the case of the tempdb database, use one data file per core.

Installing SQL Server 2008 Enterprise Edition

As we explained above, we used a dedicated Active Directory server. We highly recommend against deploying SQL Server on any Active Directory controller. Active Directory processes add overhead that could adversely affect the performance of SQL Server.

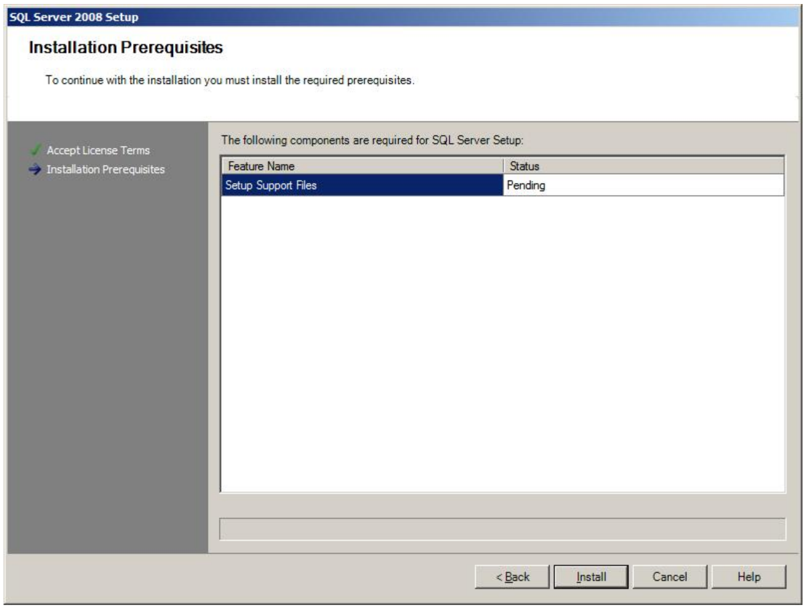
You should, however, deploy SQL Server on member servers in an Active Directory domain. Do not make the SQL Server service domain accounts members of the Domain Administrators group. In fact, grant only the necessary rights on the local server to the SQL Server service account as part of your pre-installation checklist. The SQL Server installation software creates the local groups it needs for its security purposes. For more information on the Windows Domain accounts you need for SQL Server service accounts, see SQL Server Books Online at http://msdn2.microsoft.com/en-us/library/ms143504(SQL.100).aspx#Review_NT_rights.

This section provides an overview of the SQL Server 2008 Enterprise Edition installation process. We include approximate wait times for each step. Appendix D provides detailed installation instructions.

NOTE: We installed SQL Server 2008 Enterprise Edition using an ISO image we burned to a DVD. Microsoft's download page for the February CTP of SQL Server 2008 Enterprise Edition offers two options: a self-extracting executable, which enables you to install without a DVD burner, and the DVD ISO image. Microsoft did not recommend one method over the other. We used the DVD approach because we have found it to be convenient when installing multiple times on multiple systems. If you use the self-extracting executable, your times might vary.

Plan on at least 30 minutes for installing SQL Server 2008 Enterprise Edition on the Dell PowerEdge 2950 server. Each step below takes at least 1 minute. We provide the approximate amount of time each step takes in parentheses. These times exclude data entry time.

1. Insert the SQL Server 2008 Enterprise Edition DVD into the DVD drive. Click Server components, Tools, Books Online, and Samples; accept the license agreement; and install the prerequisites (See Figure 11). (less than 1 minute)



SQL Server 2008 Setup

Installation Prerequisites

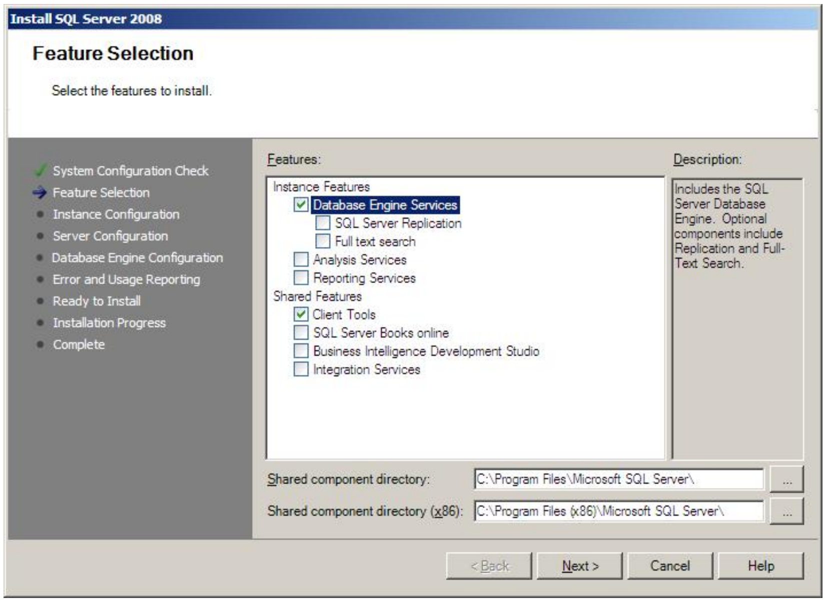
To continue with the installation you must install the required prerequisites.

The following components are required for SQL Server Setup:

Feature Name	Status
Setup Support Files	Pending

Figure 11. Installing SQL Server 2008 prerequisites.

2. Click New Installation to launch the Installing SQL Server 2008 Wizard. The system configuration check completes almost instantly. Select the features you need to install. For a basic initial installation, install the Database Engine and Client Tools (see Figure 12). You can install additional components later as necessary.


Install SQL Server 2008

Feature Selection

Select the features to install.

Features:

Instance Features
- ☑ Database Engine Services
  - □ SQL Server Replication
  - □ Full text search
- □ Analysis Services
- □ Reporting Services

Shared Features
- ☑ Client Tools
- □ SQL Server Books online
- □ Business Intelligence Development Studio
- □ Integration Services

Description:

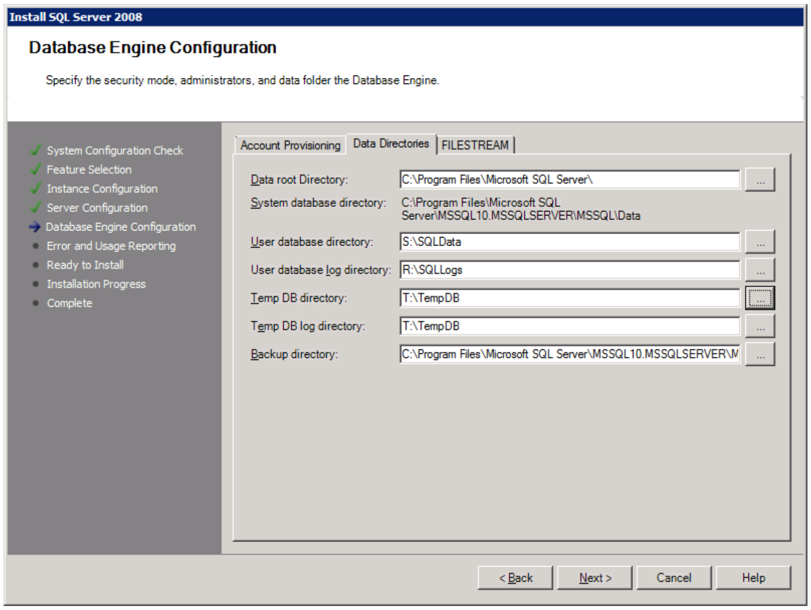
Includes the SQL Server Database Engine. Optional components include Replication and Full-Text Search.

Shared component directory: C:\Program Files\Microsoft SQL Server\

Shared component directory (x86): C:\Program Files (x86)\Microsoft SQL Server\

Figure 12. Selecting SQL Server 2008 features.

3. On the following screens, fill in the account information and data file locations (Figure 13). Install. At the confirmation screen, click Next. (7 minutes)



Database Engine Configuration

Specify the security mode, administrators, and data folder the Database Engine.

Data root Directory: C:\Program Files\Microsoft SQL Server\

System database directory: C:\Program Files\Microsoft SQL Server\MSSQL10.MSSQLSERVER\MSSQL\Data

User database directory: S:\SQLData

User database log directory: R:\SQLLogs

Temp DB directory: T:\TempDB

Temp DB log directory: T:\TempDB

Backup directory: C:\Program Files\Microsoft SQL Server\MSSQL10.MSSQLSERVER\M

Figure 13. Setting file locations.

4. At the Complete screen, note the Summary log file location, and click Close to complete the install.

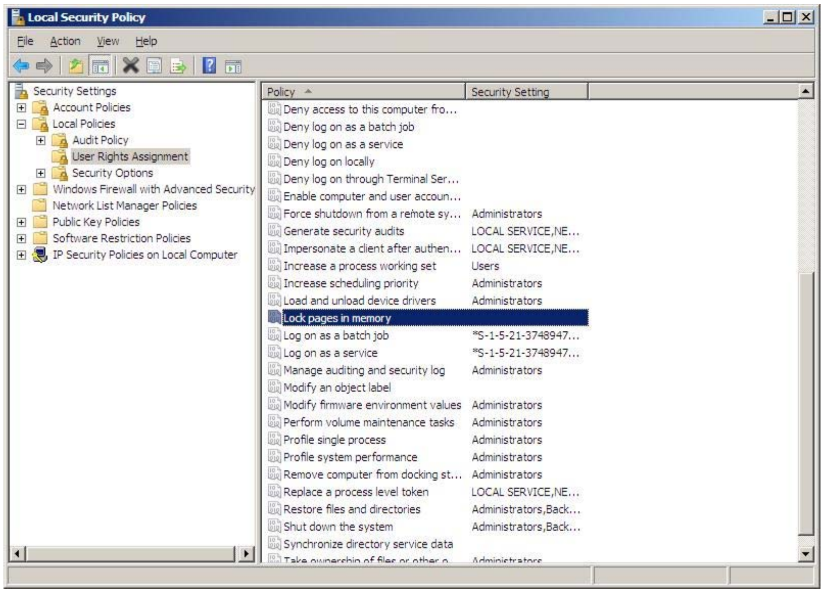
Because we were installing a Community Technology Preview (CTP) release of SQL Server 2008 Enterprise Edition, a preliminary version Microsoft releases to the technical community, there were no updates to download. When you install the release version of SQL Server 2008 Enterprise Edition, after the installation process completes, check the Microsoft Downloads Center (www.microsoft.com/downloads) for the latest SQL Server patches, updates, and service packs.

BEST PRACTICE: After you complete the SQL Server installation, give the SQL Server service account the right to prevent the operating system from paging its memory to disk. SQL Server dynamically allocates and deallocates memory to relieve memory pressure and swapping. Another process, however, can always request a substantial amount of memory and thus cause the OS to swap some of SQL Server's memory to disk before SQL Server has the chance to react. This setting prevents that.

Windows Server 2008 Enterprise Edition provides a setting that allows SQL Server to retain its data in physical memory instead of swapping the data to virtual memory, the page file, or disk. To enable this setting, give the account running the SQL Server 2008

Enterprise Edition service the right to “lock pages in memory.” To do so, follow these steps:

1. Click Start.
2. Go to Administrative tools | Local Security Policy.
3. Click Local Policies, and then click User Rights Assignment.
4. Scroll down to Lock pages in memory, and double-click (See Figure 14).
5. Click Add User or Group, and enter the SQL Server service account name.



Policy	Security Setting
Deny access to this computer fro...	
Deny log on as a batch job	
Deny log on as a service	
Deny log on locally	
Deny log on through Terminal Ser...	
Enable computer and user accoun...	
Force shutdown from a remote sy...	Administrators
Generate security audits	LOCAL SERVICE,NE...
Impersonate a client after authen...	LOCAL SERVICE,NE...
Increase a process working set	Users
Increase scheduling priority	Administrators
Load and unload device drivers	Administrators
Lock pages in memory	
Log on as a batch job	*S-1-5-21-3748947...
Log on as a service	*S-1-5-21-3748947...
Manage auditing and security log	Administrators
Modify an object label	
Modify firmware environment values	Administrators
Perform volume maintenance tasks	Administrators
Profile single process	Administrators
Profile system performance	Administrators
Remove computer from docking st...	Administrators
Replace a process level token	LOCAL SERVICE,NE...
Restore files and directories	Administrators,Back...
Shut down the system	Administrators,Back...
Synchronize directory service data	
Take ownership of files or other o...	Administrators

Figure 14. Selecting the Lock pages in memory setting.

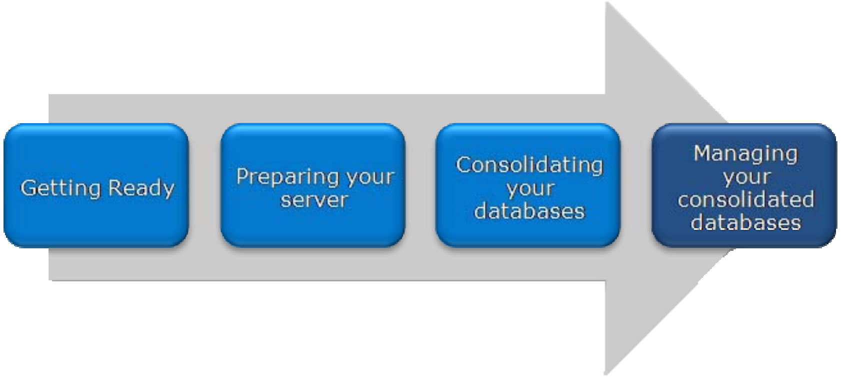
BEST PRACTICES: Leave the Minimum server memory and the Maximum server memory at their defaults of 0 and 2147483647, respectively. Doing so allows SQL Server 2008 Enterprise Edition to use as much memory as the system makes available. If you must change these settings, be sure that the sum of the maximum memory settings across all processes is less than the amount of physical RAM available.

Enable instant file initialization. The default behavior is to initialize the storage with zeros whenever SQL Server creates a data file or allocates an extent to grow a data file. Zero-filling storage can be very time consuming. When you enable instant file initialization, the system does not initialize storage it allocates. Instead, the storage remains uninitialized until SQL Server writes data to it. According to Microsoft, instant file initialization can greatly reduce the performance impact of file growth (see Table 3 at http://www.microsoft.com/technet/prodtechnol/sql/2005/physdbstor.mspx).

To enable instant file initialization, you must give the SE_MANAGE_VOLUME_NAME permission to the Windows account under which you run the SQL Server 2008 Enterprise Edition service. To do so, allow the account to “Perform volume maintenance tasks” with the following steps:

1. Click Start.
2. Go to Administrative tools | Local Security Policy.
3. Click User Rights assignment.
4. Scroll down to Perform volume maintenance tasks, and double-click.
5. Click Add User or Group, and enter the account name.

Consolidating your databases


Getting Ready
Preparing your server
Consolidating your databases
Managing your consolidated databases

Moving databases from SQL Server 2000 to SQL Server 2008

In this section, we provide an overview of the processes involved in moving your database from SQL Server 2000 to SQL Server 2008 on the Dell PowerEdge 2950. We provide detailed instructions in Appendix E.

We performed all SQL Server 2000 administration using Query Analyzer and Enterprise Manager, the two main tools in the SQL Server 2000 environment. Likewise, we performed all administration for the SQL Server 2008 installation using SQL Server Management Studio, the main administration interface for SQL Server 2008.

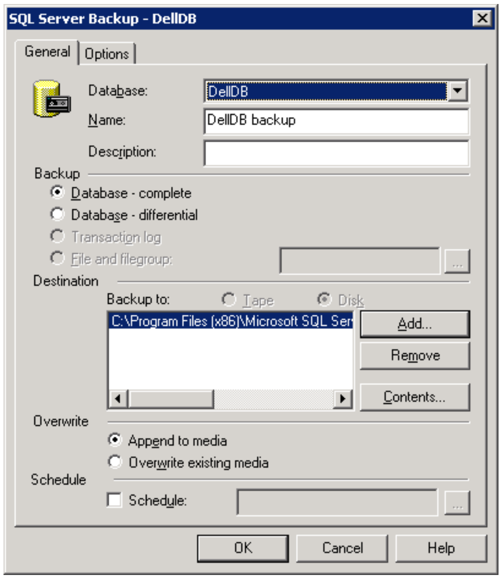
NOTE: You can move databases between instances and versions of SQL Server by various methods, including detach/attach, backup/restore, and the copy database wizard. We chose to use the backup/restore method.

We highly advise you to run the SQL Server Upgrade Advisor before upgrading your databases. For more information about the Upgrade Advisor, see http://msdn.microsoft.com/en-us/library/ms144256(SQL.100).aspx. Additional information is also available in “Migrating SQL Server 2000 to SQL Server 2008 on Windows Server 2008 Enterprise on Dell PowerEdge R900” at http://www.principledtechnologies.com/Clients/Reports/Dell/sql2000_to_sql2008_ws2008.pdf.

Back up your SQL Server 2000 database

Using the SQL Server 2000 server, take the following steps to back up your database. Backup times vary depending on the size and complexity of the database.

1. To keep users from issuing updates during the migration process, you can either set the database to read-only mode or set the access property to SINGLE_USER. Both choices immediately sever all user connections. See Appendix E for details.
2. Perform a full backup of your SQL Server 2000 database (see Figure 15). See Appendix E for details.



Database: DellDB

Name: DellDB backup

Description:

Backup

● Database - complete
○ Database - differential
○ Transaction log
○ File and filegroup:

Destination

Backup to: ○ Tape ● Disk

C:\Program Files (x86)\Microsoft SQL Ser

Overwrite

● Append to media
○ Overwrite existing media

Schedule

□ Schedule:

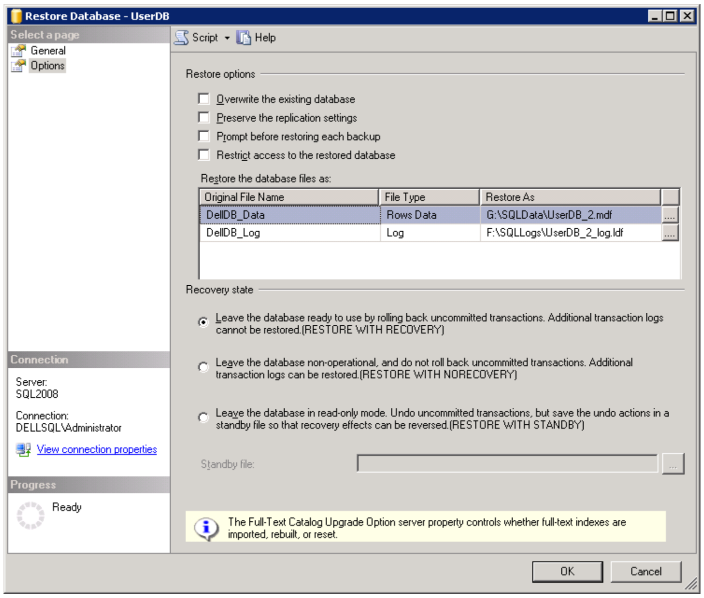
Figure 15. SQL Server 2000 backup configuration.

NOTE: If the time a full backup requires is unacceptable given your consolidation maintenance schedule, you can make the full backup at a previously scheduled time, and at this point in the process take only a differential or transaction log backup. If the transaction log size is small relative to the data file size, this approach can decrease backup and restore time significantly. Be certain, however, to keep your backup chain intact if you choose this approach.

Restore database to SQL Server 2008 server

On the SQL Server 2008 server, take the following steps:

1. Open SQL Server Management Studio, connect to the SQL Server 2008 instance, and restore the database (See Figure 16). See Appendix E for details.



Restore options

☐ Overwrite the existing database
☐ Preserve the replication settings
☐ Prompt before restoring each backup
☐ Restrict access to the restored database

Restore the database files as:

Original File Name	File Type	Restore As
DellDB_Data	Rows Data	G:\SQLData\UserDB_2.mdf
DellDB_Log	Log	F:\SQLLogs\UserDB_2_log.ldf

Recovery state

● Leave the database ready to use by rolling back uncommitted transactions. Additional transaction logs cannot be restored.(RESTORE WITH RECOVERY)

○ Leave the database non-operational, and do not roll back uncommitted transactions. Additional transaction logs can be restored.(RESTORE WITH NORECOVERY)

○ Leave the database in read-only mode. Undo uncommitted transactions, but save the undo actions in a standby file so that recovery effects can be reversed.(RESTORE WITH STANDBY)

Standby file:

The Full-Text Catalog Upgrade Option server property controls whether full-text indexes are imported, rebuilt, or reset.

Connection

Server:
SQL2008

Connection:
DELLSQL\Administrator

View connection properties

Progress

Ready

Figure 16. SQL Server 2008 restore configuration.

NOTE: It is important at this point to name your new database the same name as your SQL Server 2000 database. Changing the name may break applications that refer to the database by name.

2. While in SQL Server Management Studio, reset the database access property to multi-user, and set the compatibility level to SQL 2008 (level 100). See Appendix E for details.

Moving databases from SQL Server 2005 to SQL Server 2008

In this section, we provide an overview of the processes involved in moving your database from SQL Server 2005 to the consolidated SQL Server 2008 on the Dell PowerEdge 2950. We provide detailed instructions in Appendix F. This procedure is very similar to the SQL Server 2000 procedure, but because the management interface changed in SQL Server 2005, we also discuss this procedure.

We used SQL Server Management Studio on both the source instance and the destination server, the PowerEdge 2950. Though the 2005 and 2008 versions of Management Studio differ in some ways, none of those changes affects this procedure.

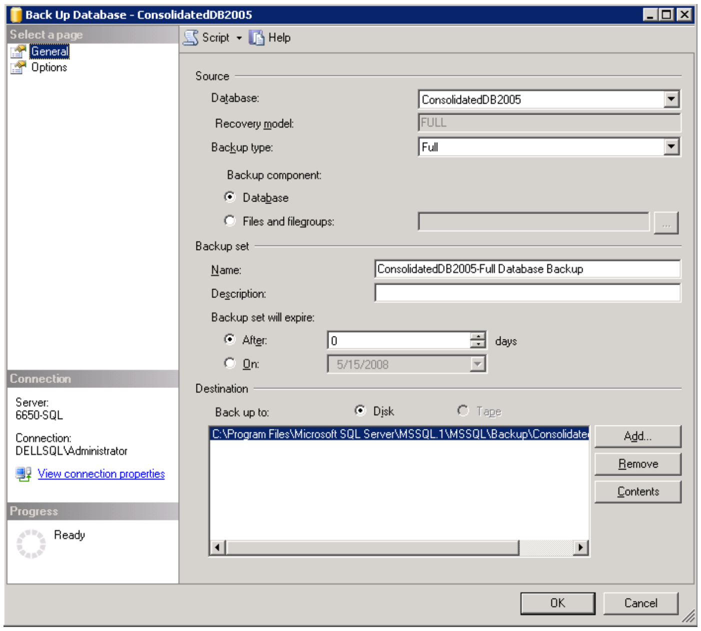
NOTE: You can move databases between instances and versions of SQL Server by various methods, including detach/attach, backup/restore, and the copy database wizard. We chose to use the backup/restore method.

As we mentioned in the Moving databases from SQL Server 2000 to SQL Server 2008 section, we highly advise you to run the Upgrade Advisor before upgrading your databases.

Back up your SQL Server 2005 database.

Using the SQL Server 2005 server, take the following steps to back up your database. Backup times vary depending on the size and complexity of the database.

1. To keep users from issuing updates during the migration process, you can either set the database to read-only mode, or set the access property to SINGLE_USER. Both choices immediately sever all user connections. See Appendix F for details on setting the database to single-user mode. We chose to set the database to SINGLE_USER mode. Be aware that in SQL Server Management Studio, it may be necessary to close the Object Explorer Details pane, the Summary pane, and any other parts of the user interface (UI) that you may have active. These panes open database connections to the relevant database and so may cause your SINGLE_USER operation to fail.
2. Perform a full backup of your SQL Server 2005 database (see Figure 17). See Appendix F for details.



Back Up Database - ConsolidatedDB2005

Select a page: General, Options

Source
Database: ConsolidatedDB2005
Recovery model: FULL
Backup type: Full
Backup component:
● Database
○ Files and filegroups:

Backup set
Name: ConsolidatedDB2005-Full Database Backup
Description:
Backup set will expire:
● After: 0 days
○ On: 5/15/2008

Destination
Back up to: ● Disk ○ Tape
C:\Program Files\Microsoft SQL Server\MSSQL.1\MSSQL\Backup\Consolidate

Connection
Server:
6650-SQL
Connection:
DELLSQL\Administrator

Progress
Ready

Figure 17. SQL Server 2005 backup configuration.

NOTE: If the time a full backup requires is unacceptable given your migration maintenance schedule, you can make the full backup at a previously scheduled time, and at this point of the migration only take a differential or transaction log backup. If the transaction log size is small relative to the data file size, this can decrease migration time significantly. Be certain, however, to keep your backup chain intact if you choose this approach.

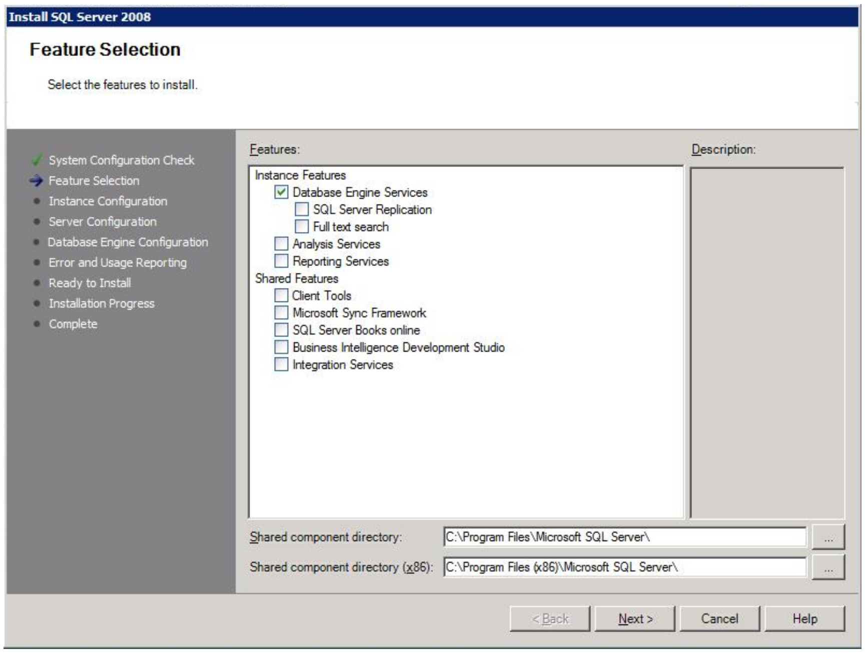
Restore database to SQL Server 2008 server

Now, simply restore your database to the SQL Server 2008 target consolidation server, using the same steps that you used to restore your SQL Server 2000 database. Unless your requirements call for retaining compatibility levels of 80 or 90, be sure and set the compatibility level to 100 to take advantage of all of the features of SQL Server 2008.

Creating a second SQL Server instance on your server

While there can be only one default instance, SQL Server 2008 allows you to have multiple named instances. These instances function largely independently of one another; each has its own security contexts, collation settings, etc. (For more information, see http://msdn.microsoft.com/en-us/library/ms143531(SQL.100).aspx.)

To install a named instance, you basically follow the same procedures as installing a default instance, with the exception of a few configuration changes during the process. First, on the Feature Selection screen, if you are installing only a second instance of the database engine, you need only check the Database Engine checkbox (see Figure 18). The instances share the workstation components, so you don’t need to install those components a second time.



Feature Selection

Select the features to install.

Features:

Instance Features
- ☑ Database Engine Services
  - □ SQL Server Replication
  - □ Full text search
- □ Analysis Services
- □ Reporting Services

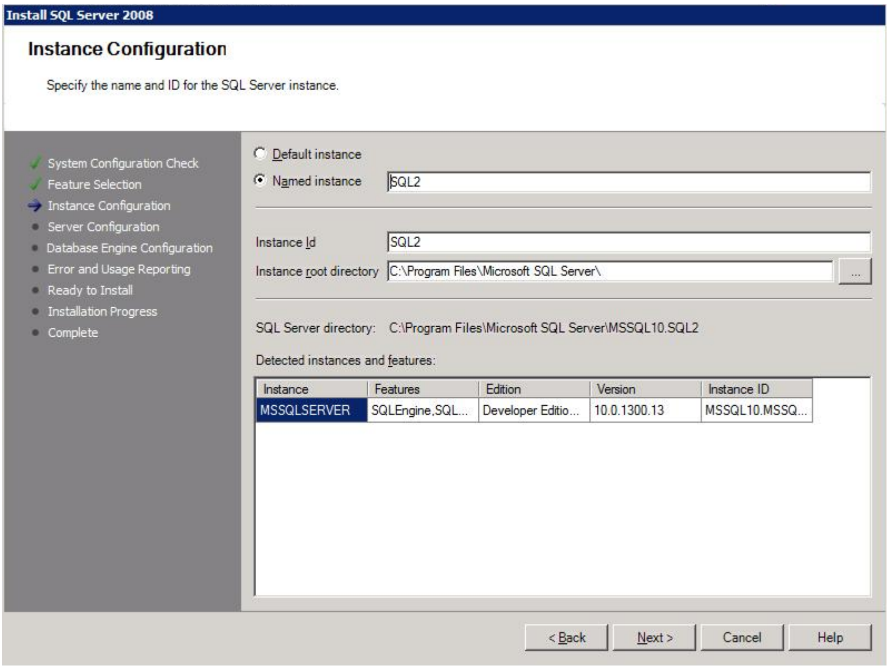
Shared Features
- □ Client Tools
- □ Microsoft Sync Framework
- □ SQL Server Books online
- □ Business Intelligence Development Studio
- □ Integration Services

Shared component directory: C:\Program Files\Microsoft SQL Server\

Shared component directory (x86): C:\Program Files (x86)\Microsoft SQL Server\

Figure 18. Choosing components to install when installing a named SQL Server instance.

The second and more important change in the installation procedure for a named instance is on the Instance Configuration screen. Here you must choose the Named Instance option and supply a name for the instance. Organizations typically have naming conventions for instances that make instance management easier, so follow those rules when naming this instance. Figure 19 shows an example of how to install and configure a named instance “SQL2”.



Install SQL Server 2008

Instance Configuration

Specify the name and ID for the SQL Server instance.

- ✓ System Configuration Check
- ✓ Feature Selection
- → Instance Configuration
- Server Configuration
- Database Engine Configuration
- Error and Usage Reporting
- Ready to Install
- Installation Progress
- Complete

○ Default instance

● Named instance: SQL2

Instance Id: SQL2

Instance root directory: C:\Program Files\Microsoft SQL Server\

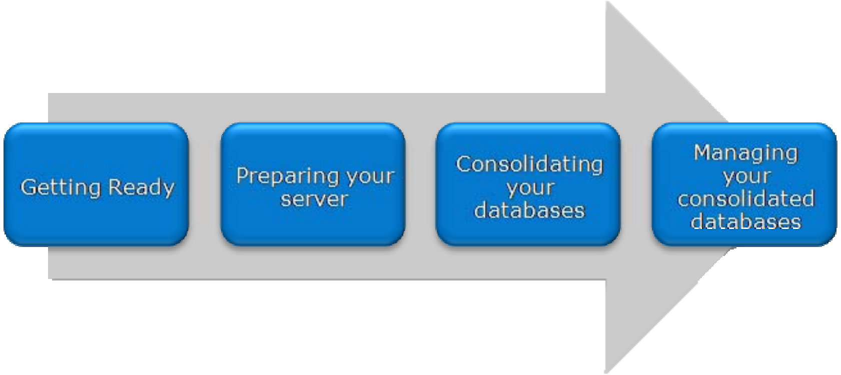
SQL Server directory: C:\Program Files\Microsoft SQL Server\MSSQL10.SQL2

Detected instances and features:

Instance	Features	Edition	Version	Instance ID
MSSQLSERVER	SQLEngine,SQL...	Developer Editio...	10.0.1300.13	MSSQL10.MSSQ...

Figure 19. Specifying configuration details for the new named instance.

Managing your consolidated databases


Getting Ready
Preparing your server
Consolidating your databases
Managing your consolidated databases

After you have completed your consolidation, you will typically need to perform some post-consolidation and management tasks. Your specific list of tasks will depend heavily on your pre-consolidation research and planning. In this section, we briefly discuss a couple of the most common tasks.

We also discuss tools, such as the Resource Governor, that will help you manage your new consolidated environment.

Transferring logins

Windows and SQL Server logins

There are two methods of authenticating to SQL Server 2008: Windows logins and SQL Server logins. Windows logins exist in the context of an Active Directory domain, and you can assign those logins rights to SQL Server resources. You create and manage SQL Server logins, however, within SQL Server. The processes for extracting login information and creating the transferred login entities on the consolidation server are very similar for both Windows authenticated logins and SQL Server authenticated logins. You must, however, take some extra steps to ensure a smooth migration for SQL Server logins.

NOTE: Take special care to avoid login issues for users when consolidating your databases. Typically these issues arise when a user had multiple SQL Server logins in the previous environment. In that situation, a user may have used different passwords for different databases, but on the consolidated server, he or she has only one password. You should note other issues in your planning documentation, such as permissions for multiple databases, default database, and language settings.

Below we describe how to generate Transact SQL (TSQL) scripts that you can use to recreate both Windows authenticated logins and SQL Server authenticated logins on the new SQL Server 2008 server.

See Appendix G for details on transferring Windows logins and Appendix H for details on transferring SQL Server logins.

To transfer logins, take the following steps on the SQL Server 2008 server. Note that steps 1 through 5 apply to both Windows logins and SQL Server logins.

1. In SQL Server Management Studio, in the Object Explorer pane, connect to either your SQL Server 2000 server or your SQL Server 2005 server, and also connect to your SQL Server 2008 server. Be sure to have the Object Explorer Details tab open (View | Object Explorer Details).
2. Expand the treeview of the SQL Server 2000 or 2005 server, browse to the Security tab, and click the Logins node. In the Object Explorer Details, a list of all logins on the SQL Server 2000 or 2005 server appears.
3. If necessary, use the sorting and filtering options in the Object Explorer Details tab, and note which logins you would like to migrate. Select them by clicking and using the standard Windows controls (Ctrl key, Shift key, etc.) to select multiple logins.
4. Right-click the selected logins, and select Script Login As | Create To | New Query Window. If you have not already connected to your new SQL Server 2008 server, do so now by right-clicking and selecting Change Connection.
5. Execute the script on your SQL Server 2008 server to create the logins. If you are transferring Windows-based logins, the process is complete.

For SQL Server logins, the script you execute in step 5 above creates the login, marks it as disabled, assigns it a random password, and does not map the login to the database user. To avoid having a database user that is “orphaned” from the login, use the provided stored procedure called sp_change_users_login to view orphaned users and to map the user to the login. We provide an example of this process in Appendix H.

Setting up Resource Governor

It has always been important in a consolidated SQL Server environment to prevent one user, set of users, or application from performing queries or other functions that damage performance for the rest of the environment. In past versions of SQL Server, database administrators had to come up with creative methods of detecting these “runaway” operations and managing them. Also, administrators typically performed resource-intensive maintenance operations after normal business hours so those activities did not affect the user population. One of SQL Server 2008’s new features, Resource Governor, offers help with these types of issues.

The Resource Governor lets you cap the CPU and memory use for different user groups. You can allocate minimum and maximum amounts of CPU and memory to different roles, and you can also programmatically configure how SQL Server classifies users into those roles.

Three concepts are central to understanding the Resource Governor functionality: resource pools, workload groups, and classification.

Resource pools

Resource pools are pools of physical resources available to SQL Server. By default, SQL Server 2008 creates two resource pools: default and internal. SQL Server 2008 reserves the internal resource pool and uses it for system tasks; you cannot modify the internal resource pool settings or map user-defined workload groups to it. By default, SQL Server 2008 does not restrict resource consumption by the internal resource pool. If SQL Server needs resources for vital system tasks, then the internal resource pool receives highest priority and thus may pressure other pools. SQL Server 2008 supports user-defined resource pools, the creation of which we cover below.

Workload groups

Workload groups are essentially containers that group sessions into categories. Each workgroup maps to a specific resource pool. When SQL Server initiates a session, it uses a classification scheme of the DBA’s design to assign the connection to a workload group.

Classification

Simply put, classification is SQL Server 2008’s method of routing session requests to the appropriate workload groups. You implement classification by writing a custom user-defined function (UDF) that categorizes connections or sessions. You may use the following system functions inside your classification UDF:

- HOST_NAME()
- APP_NAME()
- SUSER_NAME()
- SUSER_SNAME()
- IS_SRVROLEMEMBER()
- IS_MEMBER()

For more information, see the TSQL reference at SQL Server Books Online at http://msdn.microsoft.com/en-us/library/bb510741(SQL.100).aspx.

To configure a simple scenario where you create two workload groups and two resource pools, and map them to one another, follow the overview below.

1. Create a UDF in the master database that will classify your session requests into the appropriate workload groups. You will use this UDF later in your configuration of Resource Governor. For details on classification functions, see SQL Server Books Online at http://msdn.microsoft.com/en-us/library/bb933865(SQL.100).aspx.

2. Open SQL Server Management Studio, and browse to the Management node in Object Explorer. Right-click Resource Governor, and select New Resource Pool (see Figure 20).

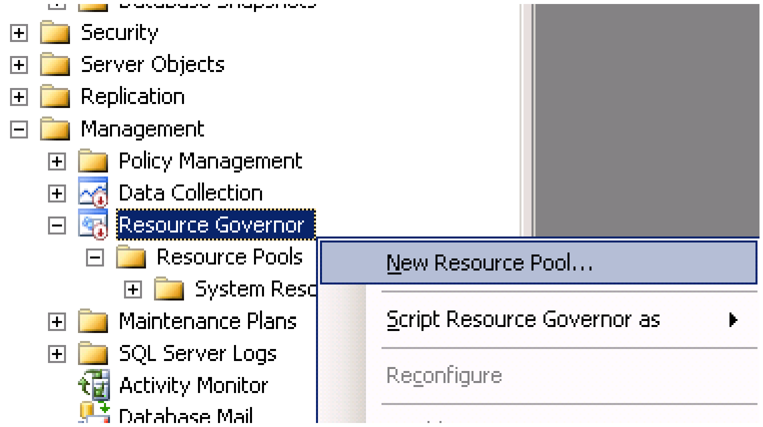
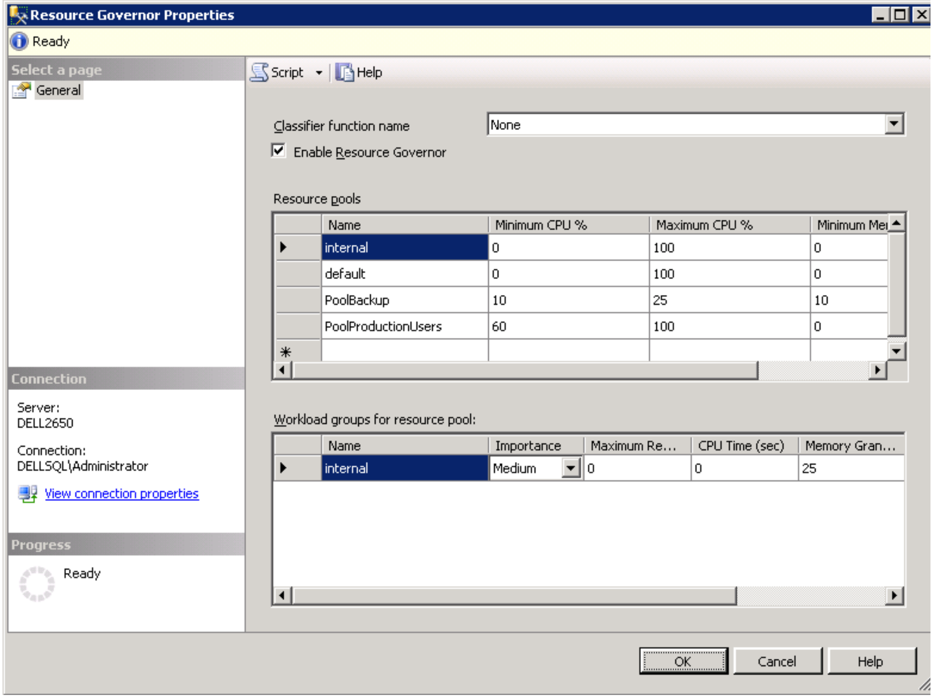


Figure 20. Creating a new resource pool via SQL Server Management Studio.

3. Choose to enable Resource Governor, choose your classification UDF, add your resource pools, and configure the minimum and maximum CPU and memory.

4. Add the corresponding workload group(s) and set an importance level (see Figure 21).



Classifier function name: None

☑ Enable Resource Governor

Resource pools

	Name	Minimum CPU %	Maximum CPU %	Minimum Me
▶	internal	0	100	0
	default	0	100	0
	PoolBackup	10	25	10
	PoolProductionUsers	60	100	0
*				

Workload groups for resource pool:

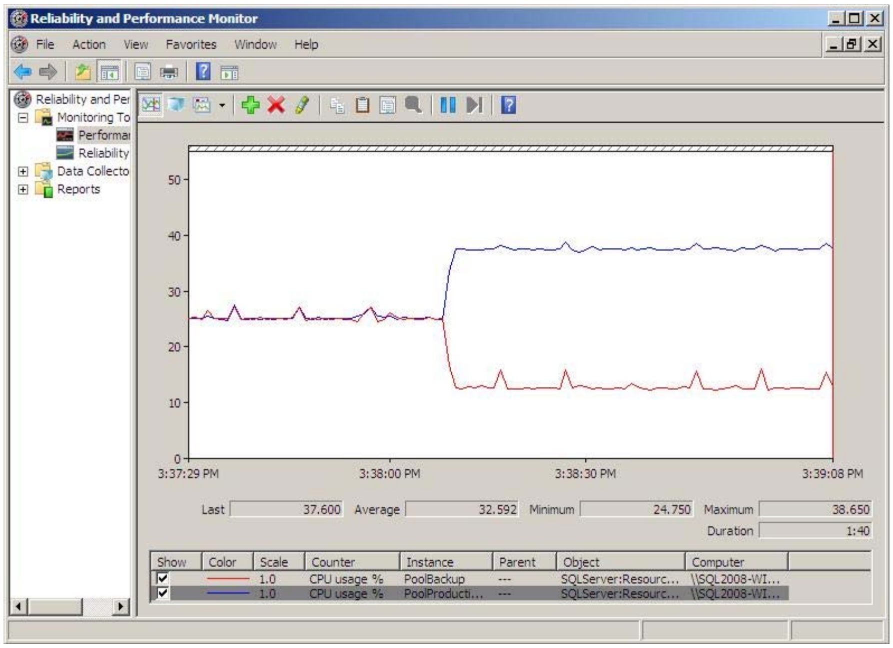
	Name	Importance	Maximum Re...	CPU Time (sec)	Memory Gran...
▶	internal	Medium	0	0	25

Server:
DELL2650

Connection:
DELLSQL\Administrator

Figure 21. Resource pool and workload group configuration in SQL Server Management Studio.

The previous example showed how to create the resource pools and workload groups necessary for Resource Governor to function. In Figure 22, a chart from the Reliability and Performance Monitor shows both groups executing a CPU-intensive query. The first part of the graph shows resource use before we enabled Resource Governor; at that time, both groups were competing for CPU cycles. Once we enabled Resource Governor, it allocated the pool of CPU resources, causing the workload priorities to diverge. This example demonstrates how quickly and efficiently Resource Governor can manage priorities.



Last 37.600 Average 32.592 Minimum 24.750 Maximum 38.650
Duration 1:40

Show	Color	Scale	Counter	Instance	Parent	Object	Computer
☑	🟥	1.0	CPU usage %	PoolBackup	---	SQLServer:Resourc...	\\SQL2008-WI...
☑	🟦	1.0	CPU usage %	PoolProducti...	---	SQLServer:Resourc...	\\SQL2008-WI...

Figure 22. Performance Monitor graph of Resource Governor's effect on two configured workload groups.

Auditing, tracking data changes, and encrypting data

SQL Server 2008 offer new features relating to auditing, tracking data changes, and encrypting data to facilitate compliance with auditing and regulatory requirements.

Auditing. Increasingly stringent regulatory requirements make auditing server operations across the enterprise more essential than ever. In SQL Server 2005, Microsoft introduced the data definition language (DDL) trigger feature, which allowed DBAs to design triggers around certain SQL Server actions, such as table creation, login creation, and so on. In SQL Server 2008, you can create new audit objects and associate audit specifications with these audit objects. Administrators can stream the audit to a file, security log, or application log. You can also configure the audit to queue its logging to lower its impact on performance. For more information, see SQL Server Books Online at http://msdn.microsoft.com/en-us/library/cc280386(SQL.100).aspx

Tracking data changes. In past versions of SQL Server, for tracking purposes administrators often used triggers to capture changes to certain data. This tactic created extra database triggers and tables they then had to maintain. Two new SQL Server 2008 features let data administrators see what data is changing, when it is changing, and who is changing it. The Change Tracking feature allows SQL Server to capture the fact that a row or column changed and what type of change occurred (insert, update, or delete). Change Data Capture is a separate feature that extracts not only the fact that data was changed, but the actual contents of the change. You must take special considerations before utilizing either of these features, because both have retention schedules and maintenance needs. For more information, see SQL Server Books Online at http://msdn.microsoft.com/en-us/library/bb933994(SQL.100).aspx.

Encrypting data. SQL Server 2008 introduces transparent data encryption (TDE). TDE lets you encrypt an entire database without affecting existing applications. TDE encrypts all data and keys and indexes automatically, allowing you to use the full power of SQL Server without sacrificing security. For more information, see SQL Server Books Online at http://msdn.microsoft.com/en-us/library/cc278098(SQL.100).aspx.

Performance Data Collector

In a consolidated environment, because a larger percentage of your enterprise is utilizing fewer pieces of hardware, it becomes more critical to capture and analyze performance statistics, so that you can optimize your configuration as much as possible.

SQL Server 2008 includes a data collector toolset to capture performance metrics for your servers. These tools include the data collection sets, the Management Data Warehouse, and such widely used tools of previous releases as SQL Server Profiler and the Database Engine Tuning Advisor. When configuring your Management Data Warehouse, you can choose to store these metrics on a centralized server.

As of the February 2008 CTP of SQL Server 2008, the Data Collector component installs with three System Data Collection Sets: Disk Usage, Query Statistics, and Server Activity. Figure 23 displays a sample report from the Server Activity system data collection set.

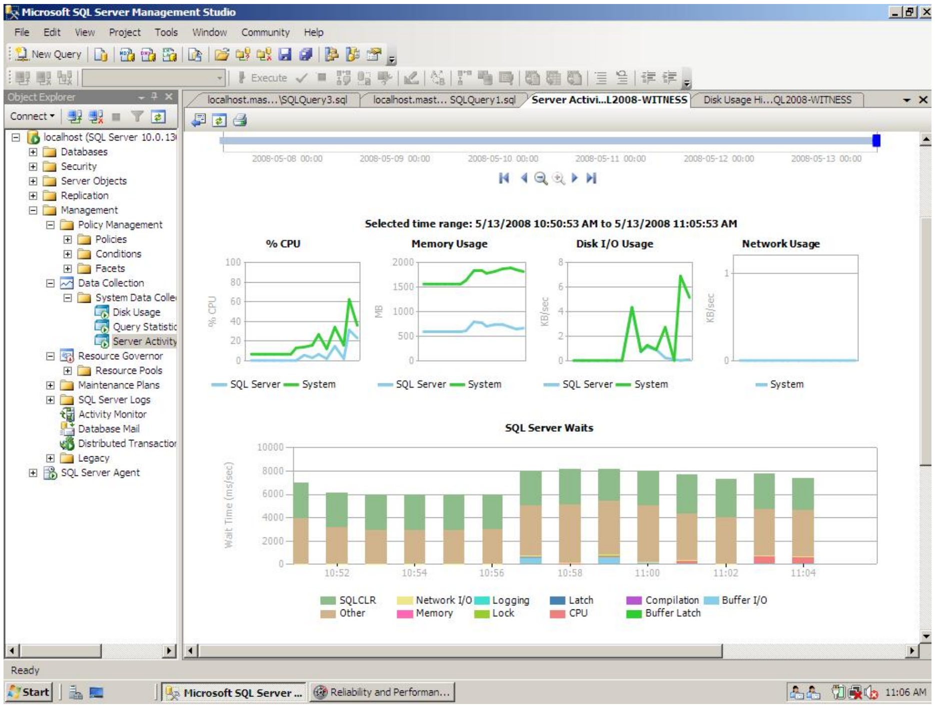


Figure 23. Sample historical report from the Server Activity data collection set.

To configure the management data warehouse and begin collection of the three default system data collection sets, follow these steps:

1. Open SQL Server Management Studio on the consolidated SQL Server 2008 server. Ensure that SQL Agent is running, because this configuration wizard creates several SQL Agent jobs and schedules. If the SQL Agent service is not running, right-click SQL Server Agent, and select Start.
2. In Object Explorer, browse to Management, then right-click Data Collection.
3. Start the Configure Management Data Warehouse Wizard, and then click Next. For the server name, select the central repository where you want to store your performance data warehouse. Then select a database, and finally select a local cache location.
4. Assign appropriate rights as you wish, then click Next, and finally click Finish.

If you wish to modify schedules, upload settings, or other configurations, right-click the specific data collection set, and select Properties.

Summing up

As this Guide has explained, the process of consolidating your databases on a Dell PowerEdge Server is relatively straightforward; you can perform a basic installation and move databases in a couple of hours. We still recommend, however, that you first invest ample time in the pre-consolidation planning phase, because doing so can help you avoid potential problems during your consolidation.

Look to upcoming guides for hard performance data that will help you size the servers you need.

Appendix A. Example server and database consolidation survey

In this appendix, we give an example survey of detailed information you may want to gather about the SQL Server instances and databases you're targeting for consolidation. While this survey is a good starting point, it may not contain every relevant detail for your particular environment.

1. Hardware level
   - a. CPU
     - i. Vendor
     - ii. Model
     - iii. Number of cores per socket
   - b. Memory
     - i. Total quantity in GB
     - ii. Memory speed
   - c. Storage
     - i. Quantity, both current and projected data growth (see database section)
     - ii. Disk speed
     - iii. Disk interface (SCSI, SAS, etc.)
2. OS level
   - a. Server name
   - b. IP address, subnet, etc.
   - c. Domain information
   - d. Operating system version, build number, and service pack
   - e. Drive letter information and layout
3. SQL instance level
   - a. Whether it's clustered, version, edition (Workgroup, Standard, Enterprise), 64-bit or 32-bit (and if 32-bit, with or without AWE), and service pack
   - b. Authentication mode (Windows only or Mixed Mode?)
   - c. Instance name (if not a default instance)
   - d. SQL port number (i.e., is it the default 1433 or another port? If a named instance, what is the port?)
   - e. Communication protocol (named pipes or TCP/IP?)
   - f. Service account and all service permission information (does SQL Agent run under a different service account?)

g. Are there any non-default master or model database objects?

h. Are there any linked server objects?

i. Are other SQL modules involved or dependent on this instance (e.g., Analysis Services, Reporting Services, etc.)?

j. Default installation directories, data directories, and log directories

k. Tempdb – highly volatile? Slightly volatile? Medium usage?

4. Database level

a. Database properties and options

i. Recovery model, auto-shrink, etc.

b. Files and filegroups

i. Size

ii. Location

c. Backups

i. How many full backups, and on what schedule?

ii. How many differential backups, and on what schedule?

iii. How many log backups, and on what schedule?

iv. Current backup window?

v. Can backup window shift to accommodate the consolidated backup window?

vi. Other backup issues

d. Is this database in a replication configuration?

e. Is this database a partner in a mirroring session (2005/2008 only)?

f. Do any SSIS/DTS packages reference this database?

g. Do any SQL Agent jobs reference this database?

h. Users/Logins

i. Number of users this database affects

ii. Would this database require a login whose name conflicts with another name on the target consolidation server? If so, you would need to create the login and map the database user using sp_change_users_login.

Appendix B. Creating the RAID disk groups

For our installation, we used four RAID groups, which we allocated as follows:

Operating system files and SQL Server executable files
SQL Server tempdb system database
SQL Server user database files
SQL Server transaction log files

We explain below how we configured these four disk groups. The directions assume you have not configured any virtual disks on your RAID controllers.

Cabling and storage

Allow at least 3 minutes to connect the PowerVault MD 1000 to the PowerEdge 2950.

1. Connect the SAS cable from the external PERC 5/E controller on the PowerEdge 2950 to the IN port on EMM 1 on the PowerVault MD1000.
2. Ensure the switch on the front of the PowerVault MD1000 is toggled upwards to indicate that you will be using unified mode on the MD1000.
3. Turn on the MD1000.

Creating the RAID groups for the PowerEdge 2950

In these instructions, we use the names and numbers of the controllers as our test setup displayed them. Thus the PERC 6/i showed as Controller 1 and the PERC 5/E showed as Controller 0. The numbering on your system might differ.

Allow at least 10 minutes to complete the creation of the RAID groups on the PowerEdge 2950.

1. Turn on the PowerEdge 2950.
2. During the boot sequence, you will see the message “Press <Ctrl><R> to Run Configuration Utility.” Press <Ctrl><R>. Note: You have only a few seconds to do this.
3. We first configure the mirrored RAID for the OS. Select the controller for the internal drives, which is the PERC 6/i, and press Enter.
4. While Controller 1 is highlighted, press F2, and select Create New VD.

5. With the RAID level field highlighted, press Enter to open the drop-down list, and select RAID 1.
6. Using the Tab key to navigate, move to the Physical Disks section, and use the space bar to select the first two drives (Drives 0 and 1). Then tab to highlight OK, and press Enter.
7. A message box appears telling you that initializing the RAID is highly recommended, but destructive. Press Enter to clear the message.
8. Now we configure the RAID 10 disk group for the SQL Server tempdb database. Highlight Controller 1 again, and press F2. Select Create New VD.
9. With the RAID level field highlighted, press Enter to open the drop-down list, and select RAID 10.
10. Using the Tab key to navigate, move to the Physical Disks section. Use the space bar to select the remaining four drives (Drives 2, 3, 4, and 5). Then tab to highlight OK, and press Enter.
11. A message box appears telling you that initializing the RAID is highly recommended, but destructive. Press Enter to clear the message.
12. Highlight Virtual Disk 0, and press F2.
13. Select Initialization, then press Enter, and then select Fast Init. When the system prompts you with any warnings, highlight OK, and press Enter.
14. Highlight Virtual Disk 1, and press F2.
15. Select Initialization, press Enter, and then select Fast Init. When the system prompts you with any warnings, highlight OK, and press Enter.
16. Press F12 to return to the main level of the configuration utility.
17. Select the controller for the drive array (in our case, the PERC 5/E), and press Enter.
18. We now configure the disk group to hold the SQL Server user databases. Highlight controller 0, press F2, and select Create New VD.
19. With the RAID level field highlighted, press Enter to open the drop-down list, and select RAID 10.
20. Using the Tab key to navigate, move to the Physical Disks section. Use the space bar to select the first ten drives (Drives 0, 1, 2, 3, 4, 5, 6, 7, 8, and 9). Then tab to highlight OK, and press Enter.
21. A message box appears telling you that initializing the RAID is highly recommended, but destructive. Make sure OK is selected, and press Enter.
22. We now configure the RAID 10 disk group to hold SQL Server transaction log files. Highlight Controller 0 once

again, and press F2. Select Create New VD, and press Enter.

23. With the RAID level field highlighted, press Enter to open the drop-down list, and select RAID 10. Using the Tab key to navigate, move to the Physical Disks section. Use the space bar to select the next four drives (Drives 10, 11, 12, and 13). (Note that the drives you allocated earlier no longer appear in the available list.) Tab to highlight OK, and press Enter.
24. A message box appears telling you that initializing the RAID is highly recommended, but destructive. Click OK.
25. To use one of the two remaining disks as a hot spare on the PowerVault MD1000, follow these steps:

    a. Press <Ctrl><N> to navigate to the Physical Disk page of the RAID setup. Arrow down to the last disk, which should have no Disk Group assigned.
    b. Press F2, and then select Make Global HS. Press Enter. If you see a message box asking if you want the global hot spare to give priority to the enclosure in which it resides, click Yes. This disk is now a global hot spare for both Disk Groups on the external storage array.
26. Press <Ctrl><P> to navigate back to the Virtual Disk Management screen. Highlight each Virtual Disk, and press F2. Select Initialization, and then select Fast Init. Accept OK when the system prompts you with a warning. After you select Fast Init, a progress bar appears briefly, and then disappears.
27. Exit the RAID utility, and reboot the server.

Appendix C. Installing Windows Server 2008 Enterprise Edition

Install Microsoft Windows Server 2008 Enterprise Edition by following these steps. We provide approximate times for each group of steps in the first step of that group.

Allow at least 60 minutes to complete the installation on the Dell PowerEdge 2950.

1. Insert the Microsoft Windows Server 2008 Enterprise Edition DVD into the DVD drive, and reboot the system. During the boot, a Windows is loading files message appears, followed by a Windows booting loading bar. (3 minutes).
2. At the Install Windows Server 2008 screen, accept the defaults of English language, English time and currency format, and US keyboard or input method by clicking Next.
3. Click Install now.
4. At the Type your product key for activation screen, enter your activation key, and click Next.
5. At the Select the edition of Windows that you purchased screen, click Windows Server 2008 Enterprise Edition (Full Installation), check the I have selected the edition of Windows that I purchased box, and click Next.
6. At the Please read the license terms screen, check the I accept the license terms checkbox, and click Next.
7. At the Which type of installation do you want? screen, for Type of install, click Custom (Advanced).
8. In the Where do you want to install Windows? screen, click Drive options (Advanced).
9. Highlight the partition, and click Next to start the installation. (20 minutes, during which Windows restarts twice)
10. When the system indicates that you must change the user’s password before you log on the first time, click OK.
11. Enter a strong password in both the New password and Confirm password fields, and click the arrow.
12. At the Your password has been changed prompt, click OK. Windows then prepares the desktop. (less than 1 minute)

13. Set the IP address, subnet mask, and domain information as per the steps below.
14. Install Windows updates. Depending on your particular environment, you may need to change the default firewall settings for Windows update to work. (4 minutes; download times vary)
15. Reboot the server. (2 minutes)

Setting IP address, subnet mask, and domain information

After rebooting, set the static IP addresses and subnet mask using the following process:

1. Click Start.
2. Right-click Network, and select Properties.
3. In the Network Sharing Center, click Manage network connections.
4. Right-click the Local Area Connection, and select Properties.
5. In the Local Area Connection Properties dialog, highlight Internet Protocol Version 4 (TCP/IPv4), and click Properties.
6. Click the radio button next to Use the following IP address.
7. Type the IP address you want the server to have, the Subnet mask, and the Preferred DNS server, and click OK.
8. Close the Network Connections window.
9. From the Start menu, right-click Computer, select Properties, and then select Advanced system settings.
10. Select the Computer Name tab.
11. Click Change.
12. Enter the server name.
13. Click the Member of Domain radio button.
14. Enter the domain name.
15. Enter the user name and password when the system prompts you to do so. (small delay, but less than 1 minute)
16. When the Computer Name/Domain Changes window appears, click OK to close it.
17. Click OK to go past the warning to restart.
18. Click OK to close the System Properties window.
19. Click Restart Now to reboot the server. (3 minutes)

Configuring data drives

Configure each of the three data (non-operating system) RAID drives on the server using the following steps:

1. Click Start.
2. Expand Administrative Tools, and click Computer management.
3. In the left panel, expand Storage, and select Disk Management. If the Initialize Disk dialog box appears, click OK.
4. Right-click the unallocated disk, and select New Simple Volume.
5. On the Welcome to the New Simple Volume Wizard screen, click Next.
6. Leave the volume size at the default, and click Next.
7. Leave the drive letter at the default, and click Next.
8. On the Format Partition screen, do the following:
    a. Leave the File System as NTFS.
    b. Set the Allocation Unit size to 64 K. (Note: Windows Server 2008 Enterprise Edition grays out the option to compress the drive.)
    c. Fill in a volume name if you want the disk to have one.
    d. Optionally, select the quick format checkbox.
9. Click Next.
10. On the Completing the New Simple Volume Wizard screen, click Finish.

Appendix D. Installing SQL Server 2008 Enterprise Edition

NOTE: We installed SQL Server 2008 Enterprise Edition using an ISO image we burned to a DVD. Microsoft's download page for the February CTP of SQL Server 2008 Enterprise Edition offers two options: a self-extracting executable, which enables you to install without a DVD burner, and the DVD ISO image. Microsoft did not recommend one method over the other. We used the DVD approach because we have found it to be convenient when installing multiple times on multiple systems. If you use the self-extracting executable, your times might vary.

Install Microsoft SQL Server 2008 Enterprise Edition on the PowerEdge 2950 server by following these steps. We provide approximate times for each group of steps in the first step of that group.

Allow at least 30 minutes to complete the installation on the server.

1. Insert the SQL Server 2008 Enterprise Edition DVD into the DVD drive.
2. At the SQL Server 2008 Start screen, under Install, click Server components, Tools, Books Online, and Samples. (less than 1 minute)
3. On the Accept License Terms screen, check I accept the license terms, and click Next.
4. Exit the .NET installation by clicking Exit.
5. On the Installation Prerequisites screen, click Install to install the required prerequisites. (2 minutes)
6. On the Installation Center screen, click New Installation to launch the Installing SQL Server 2008 Wizard. The system configuration check runs automatically.
7. Click Next to continue.
8. On the Feature Selection screen, select only those components you need for this particular installation of SQL Server 2008 Enterprise Edition. This practice reduces overhead on your Dell PowerEdge server. If you plan to run only the database engine, then select only Database Engine Services and Client Tools. The client tools are helpful for administering this server locally.
9. Click Next to continue.

10. On the Instance Configuration screen, either accept the Default instance or specify a named instance. You can have only one default instance on a server.
11. Click Next to continue.
12. On the Service accounts tab of the Server configuration screen, enter the account name and password you want to use. You must qualify the account name with the domain name: <domain>\<account>. If you want to use a single account for all SQL services, enter the account and password in the Use the same account for all section, and click Apply to all.
13. On the Collation tab of the Server configuration screen, accept the defaults unless you have a specific requirement for other collations, and click Next to continue.
14. On the Account Provisioning tab of the Database Engine Configuration screen, select whether you want to run SQL Server 2008 in mixed mode or Windows Authentication only. We recommend leveraging Active Directory security and selecting Windows Authentication. If, however, you must use SQL Server logins, you must select mixed mode. You must specify at least one SQL Server administrator. We clicked Add Current User. If you run SQL Server 2008 in mixed mode, you must enter the password for the “sa” account here.
15. On the Data Directories tab of the Database Engine Configuration screen, enter the appropriate path names. See the Configuring the drives section for help in selecting physical directory and file locations. If at all possible, separate the user database directory, user database log directory, and tempdb directory.
16. Click Next to continue.
17. On the Error and Usage Reporting screen, SQL Server has selected the following by default: Send error reports to Microsoft or your corporate report server, and Send anonymous feature usage data to Microsoft. Deselect these if you do not want to participate in Microsoft’s data collection program.
18. Click Next to continue.
19. The Ready to install screen offers one last chance to review your choices. After you do so, click Install. (7 minutes)
20. At the confirmation screen, click Next.
21. At the Complete screen, note the Summary log file location, and click Close to complete the installation.

22. After the SQL Server 2008 installation process completes, check the Microsoft Downloads Center (www.microsoft.com/downloads) for the latest SQL Server updates, patches, and service packs. Note: There were none at the time we wrote this Guide.
23. Ensure the account under which SQL Server is running has the Lock Pages in Memory permission:
    a. Select Start | Administrative Tools | Local Security Policy.
    b. Expand Local Policies, and click User Rights Assignment.
    c. Scroll down on the right to Lock Pages in Memory, right-click this entry, and select Properties.
    d. Click Add User or Group…, and then enter DOMAIN\SQLServiceUser in the object names area (where DOMAIN is your domain and SQLServiceUser is the domain user under which SQL Server is running).
    e. Click OK to exit.
    f. Restart the SQL Server service by selecting Start | All Programs | Microsoft SQL Server 2008 | Configuration Tools | SQL Server Configuration Manager. Expand SQL Server Services. Right-click the SQL Server service, and then select Restart.
24. To enable instant file initialization, you must give the SE_MANAGE_VOLUME_NAME permission to the Windows account under which you run the SQL Server 2008 service:
    a. Click Start.
    b. Go to Administrative tools | Local Security Policy.
    c. Click User Rights assignment.
    d. Scroll down to Perform volume maintenance tasks, and double-click.
    e. Click Add User or Group, and enter the account name.

Appendix E. Migrating the databases you are consolidating from a SQL Server 2000 environment

In this appendix, we give detailed instructions on a basic side-by-side migration of a user database from a SQL Server 2000 server to a SQL Server 2008 server. You can accomplish this task in a variety of ways, including using TSQL commands and automated tools. Here we discuss implementing the database migration using the graphical interface tools Microsoft provides with SQL Server 2000 and SQL Server 2008.

1. Log in to Windows on the SQL Server 2000 server as either an administrative user or a user with full SQL Server 2000 rights.
2. Start Query Analyzer by selecting Start | All Programs | Microsoft SQL Server | Query Analyzer.
3. Enter your server name, and select Windows Authentication.
4. Set the database to single-user mode to ensure no updates occur.

NOTE: Setting the database to single-user mode immediately disconnects all users, causing them to lose any work in progress. You must notify your users well in advance of setting the database to single-user mode.

5. To set the database to single-user mode, in the query window, paste the following command, where <DatabaseName> is the name of the database you are migrating:

```
USE [master]
GO
ALTER DATABASE [<DatabaseName>]
SET SINGLE_USER WITH ROLLBACK IMMEDIATE
GO
```

6. Press F5 to execute the query. It may not create any visible output.
7. Close Query Analyzer. When Query Analyzer gives you the chance to save the query, click No.
8. To back up the database and prepare for migration, open Enterprise Manager by selecting Start | All Programs | Microsoft SQL Server | Enterprise Manager.
9. In the left pane, expand Microsoft SQL Servers | SQL Server Group | (local) (Windows NT) | Databases.
10. Right-click the database, and select All Tasks | Backup Database.
11. Keep the default of complete backup.

12. Click Add… to add a backup device, then the "…" button to browse to the backup location you want to use. Select a folder, and enter a filename, such as DatabaseNameMigrationBackup.bak.
13. Click OK to close the Backup Device Location window, and then click OK to close the Select Backup Destination window.

NOTE: We created the backup in a shared local folder, to simplify moving the database later. Also, make sure that no unwanted destinations remain in the Backup to: box from previous backups. These can cause the backup to split into multiple files, which can cause problems later.

14. Optionally, on the Options tab, select the checkbox to Verify your database backup upon completion.

NOTE: This increases backup time, but checks the integrity of the backup file. We chose to verify the backup.

15. Click OK to begin the SQL Server 2000 database backup. (backup times vary based on database size and backup device configuration)
16. After the backup completes, in Enterprise Manager, under Databases, right-click the database, and select All Tasks | Take Offline. The old database is now offline and SQL Server 2000 does not allow any connections to it.
17. Log out of Windows on the SQL Server 2000 server.
18. Log in to Windows on the SQL Server 2008 server.
19. In Windows Explorer, navigate to either the shared network location or the portable storage device that you used to store your database backup.
20. Copy the backup file and paste to a local directory of your choice. (copy time varies based on network and file size)
21. Open SQL Server Management Studio by selecting Start | All Programs | Microsoft SQL Server 2008 | SQL Server Management Studio.
22. Select Database Engine for Server type, and type the server name in Server name. For Authentication, we used Windows Authentication.
23. In the Object Explorer pane, right-click Databases, and select Restore Database.
24. In the To database box, enter the name of the database.

NOTE: It is critical to use the same database name as on your SQL Server 2000 server. If the database name does not match, applications that depend on the database name may break.

25. Click the “From device” radio button, and then click “...” to browse to the local folder where you copied the backup file.
26. Click Add to add the backup file location. Browse to the appropriate folder, find the backup file, select it, and click OK. Then click OK to return to the Restore Database window.
27. Select the checkbox that now appears in the Restore column. Optionally, click Options on the left, and ensure the file locations are appropriate in the Restore As column.
28. Click OK to begin the restore. Note the progress indicator in the lower left of the Restore window. (restore time varies based on database size)
29. After the restore is complete, you must change the database access state back to multi-user and upgrade the compatibility level by performing the following steps:

    a. Right-click the database in Object Explorer, and select Properties.
    b. Click Options on the left side to access database options.
    c. Change Compatibility level to SQL Server 2008 (100).
    d. Scroll down to Restrict Access, and change to MULTI_USER.
    e. Click OK.
    f. Click Yes to agree to shut down other connections.

Appendix F. Migrating the databases you are consolidating from a SQL Server 2005 environment

In this appendix, we give detailed instructions about a basic side-by-side migration of a user database from a SQL Server 2005 server to a SQL Server 2008 server. You can accomplish this task in a variety of ways, including using TSQL commands and automated tools. Here we discuss performing the database migration using the graphical interface tools Microsoft provides with SQL Server 2005 and SQL Server 2008; the primary tool is SQL Server Management Studio. This appendix assumes that your SQL Server 2005 server is running Windows Server 2003 as the operating system.

1. Log in to Windows 2003 on the SQL Server 2005 server as either an administrative user or a user with full rights on SQL Server 2005.
2. Start SQL Server Management Studio by selecting Start | All Programs | Microsoft SQL Server 2005 | SQL Server Management Studio.
3. Enter your server name, and select Windows Authentication.
4. Set the database to single-user mode to ensure no updates occur.
5. Close any query window, object explorer details window, or summary windows that may be open. Leave Object Explorer open. If Object Explorer is not visible, you can invoke it by pressing the F8 key.
6. Expand Databases, then right-click the database in Object Explorer.
7. Select Properties.
8. Click Options on the left side of Database Properties window.
9. Scroll down to the last option on the Database Options window.
10. Change the "Restrict Access" option from Multiple to Single.
11. Click OK.

NOTE: Setting the database to single-user mode immediately disconnects all users, causing them to lose any work in progress. You must notify your users well in advance of setting the database to single-user mode.

12. To back up the database and prepare for migration, browse to the relevant database in Object Explorer.
13. Right-click the database, and select Tasks | Backup Database.
14. Under backup type, keep the default of Full backup.
15. Configure the backup device under the Destination section of the Backup Database configuration window. Accept the defaults unless your needs dictate otherwise. Note the location (path and filename) of the backup device.

NOTE: To simplify moving the database later, we recommend creating the backup in a shared local folder. Also, make sure that no unwanted destinations remain in the Backup to: box from previous backups. Such destinations can cause the backup to split into multiple files, which can cause problems later.

16. Optionally, on the Options tab, select the checkbox to Verify your database backup upon completion.

NOTE: This option increases backup time but checks the integrity of the backup file. We chose to verify the backup.

17. Click OK to begin the SQL Server 2005 database backup. (backup times vary with database size and backup device configuration)
18. After the backup completes, in SQL Server Management Studio's Object Explorer, under databases, right-click the database, and select Tasks | Take Offline. The old database is now offline, so SQL Server 2005 does not allow any connections to it.
19. Log out of Windows on the SQL Server 2005 server.
20. Log in to Windows on the SQL Server 2008 server.
21. In Windows Explorer, navigate to either the shared network location or the portable storage device that you used to store your database backup.
22. Copy the backup file, and paste it to a local directory of your choice. (copy time varies with network speed and file size)

23. Open SQL Server Management Studio by selecting Start | All Programs | Microsoft SQL Server 2008 | SQL Server Management Studio.
24. Select Database Engine for Server type, and type the server name in Server name. For Authentication, we used Windows Authentication.
25. In the Object Explorer pane, right-click Databases, and select Restore Database.
26. In the To database box, enter the name of the database.

NOTE: It is critical to use the same database name as on your SQL Server 2005 server. If the database name does not match, applications that depend on the database name may break.

27. Click the From device radio button, and then click "…" to browse to the local folder where you copied the backup file.
28. Click Add to add the backup file location. Browse to the appropriate folder, find the backup file, select it, and click OK. Then click OK to return to the Restore Database window.
29. Select the checkbox that now appears in the Restore column. Optionally, click Options on the left, and ensure the file locations are appropriate in the Restore As column.
30. Click OK to begin the restore. Note the progress indicator in the lower left of the Restore window. (restore time varies with database size and server and disk subsystem speed)
31. After the restore is complete, you must change the database access state back to multi-user and upgrade the compatibility level by performing the following steps:
    a. Right-click the database in Object Explorer, and select Properties.
    b. Click Options on the left side to access database options.
    c. Change Compatibility level to SQL Server 2008 (100).
    d. Scroll down to Restrict Access, and change to MULTI_USER.
    e. Click OK.
    f. Click Yes to agree to shut down other connections.

Appendix G. Transferring Windows logins via SQL Server Management Studio

In this appendix, we provide detailed instructions for migrating transferring Windows-based logins from a SQL Server 2000 or 2005 server to SQL Server 2008 in an Active Directory environment. For information about transferring logins when using SQL Server authentication, see Appendix H.

1. On the SQL Server 2008 server, log in to Windows, then open SQL Server Management Studio by selecting Start | All Programs | Microsoft SQL Server 2008 | SQL Server Management Studio.
2. On the Connect to Server screen, select Database Engine for the server type, enter the computer name of your SQL Server 2000 or 2005 server, and click Connect.
3. Select View | Object Explorer Details to bring up the Object Explorer Details tab.
4. In the Object Explorer pane, expand the Security folder, and then highlight the Logins folder.
5. Add the Login Type column to the display by right-clicking the column-header area in the Object Explorer Details pane and selecting Login Type. Now Windows Logins appear as Login Type 0. Optionally, use the column headers to sort by Default Database, or use the filter button to find only logins that you are interested in migrating.
6. Using the standard Windows multi-select key combinations (Ctrl key or Shift key), highlight the logins you are interested in migrating.
7. Right-click the highlighted logins, and select Script Login as | CREATE To | New Query Editor.
8. On the Connect to Database Engine screen, change the Server name to your SQL Server 2008 server, and click Connect.
9. Click Execute to run the resulting script.

Appendix H. Transferring SQL Server logins via SQL Server Management Studio

In this appendix, we give detailed instructions on transferring SQL Server logins from a SQL Server 2000 or 2005 server to SQL Server 2008. For information about transferring logins when using Windows authentication, see Appendix G.

Transferring logins

1. On the SQL Server 2008 server, log in to Windows, then open SQL Server Management Studio by selecting Start | All Programs | Microsoft SQL Server 2008 | SQL Server Management Studio.
2. On the Connect to Server screen, select Database Engine for the server type, enter the computer name of your SQL Server 2000 or 2005 server, and click Connect.
3. Select View | Object Explorer Details to bring up the Object Explorer Details tab.
4. In the Object Explorer pane, expand the Security folder, and then highlight the Logins folder.
5. Add the Login Type column to the display by right-clicking the column-header area and selecting Login Type. Now SQL Server Logins appear as Login Type 2. Optionally, use the column headers to sort by Default Database, or the Filter button to find only logins that you are interested in migrating.
6. Using the standard Windows multi-select key combinations (Ctrl key or Shift key), highlight the logins you are interested in migrating.
7. Right-click the highlighted logins, and select Script Login as | CREATE To | New Query Editor Window.
8. On the Connect to Database Engine screen, change the Server name to your SQL Server 2008 server, and click Connect.
9. Click Execute to run the resulting script.

Mapping transferred logins to database users

After transferring SQL Server logins from server to server, you must map those logins to the database users that you migrated during the restore process. This process assumes that you have already migrated the applicable database to the new server.

1. On the SQL Server 2008 server, log in to Windows. Then open SQL Server Management Studio by

selecting Start | All Programs | Microsoft SQL Server 2008 | SQL Server Management Studio.

2. On the Connect to Server screen, select Database Engine for the server type, enter the computer name of your SQL Server 2008 server, and click Connect.
3. Right-click the relevant database, and select New Query.
4. Enter the following command, and click Execute:

```
EXEC sp_change_users_login 'Report';
```

5. For each user output from the above step, map these “orphaned” users to the appropriate matching SQL Server login by taking the following steps.
6. Open a query window on the SQL Server 2008 using steps 1 to 3 above.
7. For each login you wish to map, enter the following command, and click Execute:

```
EXEC sp_change_users_login
'Auto_Fix','SQLLogin';
```

8. The above command maps the database user “SQLLogin” to the SQL Server login of “SQLLogin”. This procedure assumes that the database user and SQL Server login have the same value. Upon successful execution of this procedure, you will see output similar to the following:

```
The number of orphaned users fixed by
updating users was 1.
```

9. To reset the password and enable the account, enter the following command, and click Execute:

```
USE [master]
GO

ALTER LOGIN [SQLLogin] WITH
PASSWORD=N'Password1'
GO

ALTER LOGIN [SQLLogin] ENABLE
GO
```

10. The above commands reset the SQLLogin login to have a password of Password1 and also enable that login.

About Principled Technologies

Principled Technologies, Inc.
1007 Slater Road, Suite 250
Durham, NC, 27703
www.principledtechnologies.com

We provide industry-leading technology assessment and fact-based marketing services. We bring to every assignment extensive experience with and expertise in all aspects of technology testing and analysis, from researching new technologies, to developing new methodologies, to testing with existing and new tools.

When the assessment is complete, we know how to present the results to a broad range of target audiences. We provide our clients with the materials they need, from market-focused data to use in their own collateral to custom sales aids, such as test reports, performance assessments, and white papers. Every document reflects the results of our trusted independent analysis.

We provide customized services that focus on our clients' individual requirements. Whether the technology involves hardware, software, Web sites, or services, we offer the experience, expertise, and tools to help you assess how it will fare against its competition, its performance, whether it's ready to go to market, and its quality and reliability.

Our founders, Mark L. Van Name and Bill Catchings, have worked together in technology assessment for over 20 years. As journalists they published over a thousand articles on a wide array of technology subjects. They created and led the Ziff-Davis Benchmark Operation, which developed such industry-standard benchmarks as Ziff Davis Media's Winstone and WebBench. They founded and led eTesting Labs, and after the acquisition of that company by Lionbridge Technologies were the head and CTO of VeriTest.

Disclaimer of Warranties; Limitation of Liability:
PRINCIPLED TECHNOLOGIES, INC. HAS MADE REASONABLE EFFORTS TO ENSURE THE ACCURACY AND VALIDITY OF ITS TESTING, HOWEVER, PRINCIPLED TECHNOLOGIES, INC. SPECIFICALLY DISCLAIMS ANY WARRANTY, EXPRESSED OR IMPLIED, RELATING TO THE TEST RESULTS AND ANALYSIS, THEIR ACCURACY, COMPLETENESS OR QUALITY, INCLUDING ANY IMPLIED WARRANTY OF FITNESS FOR ANY PARTICULAR PURPOSE. ALL PERSONS OR ENTITIES RELYING ON THE RESULTS OF ANY TESTING DO SO AT THEIR OWN RISK, AND AGREE THAT PRINCIPLED TECHNOLOGIES, INC., ITS EMPLOYEES AND ITS SUBCONTRACTORS SHALL HAVE NO LIABILITY WHATSOEVER FROM ANY CLAIM OF LOSS OR DAMAGE ON ACCOUNT OF ANY ALLEGED ERROR OR DEFECT IN ANY TESTING PROCEDURE OR RESULT.

IN NO EVENT SHALL PRINCIPLED TECHNOLOGIES, INC. BE LIABLE FOR INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES IN CONNECTION WITH ITS TESTING, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. IN NO EVENT SHALL PRINCIPLED TECHNOLOGIES, INC.'S LIABILITY, INCLUDING FOR DIRECT DAMAGES, EXCEED THE AMOUNTS PAID IN CONNECTION WITH PRINCIPLED TECHNOLOGIES, INC.'S TESTING. CUSTOMER'S SOLE AND EXCLUSIVE REMEDIES ARE AS SET FORTH HEREIN.
Dell AD # 180001917