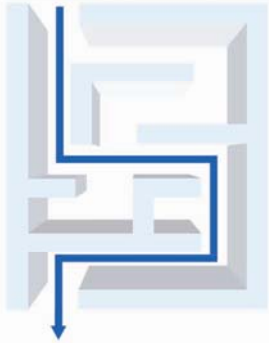


DEPLOYMENT



GUIDE

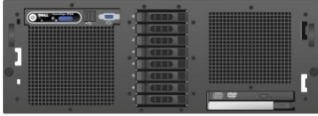
**MIGRATING SQL SERVER® 2000
TO SQL SERVER® 2008
ON
WINDOWS SERVER® 2008 ENTERPRISE
ON
DELL™ POWEREDGE™ R900**

Table of contents

Introduction	4
Scope of this Guide	4
New features in SQL Server 2008.....	4
New features of Windows Server 2008 Enterprise	5
Pre-migration considerations	7
32 bit versus 64 bit.....	7
Upgrade paths.....	9
Component considerations	9
SQL Server 2000 features not present in SQL Server 2008	10
Logins and dependencies: Pre-migration	10
Installing and setting up Windows Server 2008 Enterprise and SQL Server 2008 in this environment	12
Defining our environment	12
Configuring the RAID drives	14
Installing Windows Server 2008.....	17
Configuring the operating system drives	19
Installing SQL Server 2008.....	20
Migrating to SQL Server 2008.....	26
Using the SQL Server 2008 Upgrade Advisor to analyze databases.....	26
Implementing side-by-side migration.....	28
Post-migration considerations.....	31
Compatibility level	31
Logins and dependencies: Post-migration.....	31
Summing up	33
Appendix A. Methodology: Creating the RAID disk groups..	34
Appendix B. Methodology: Installing Windows Server 2008 Enterprise.....	37
Setting IP address, subnet mask, and domain information	38
Configuring data drives.....	39
Appendix C. Methodology: Installing SQL Server 2008	40
Appendix D. Extending available memory on 32-bit SQL Server 2008 installations – Enabling Address Windowing Extensions (AWE)	43
Appendix E. Running the SQL Server 2008 Upgrade Advisor	45
Appendix F. SQL Server 2000 to SQL Server 2008 side-by-side migration: Step by step.....	47
Appendix G. Transferring Windows logins via SQL Server Management Studio	50

Appendix H. Transferring SQL Server logins via SQL Server
Management Studio52
About Principled Technologies.....54

Introduction



Dell PowerEdge R900

The Dell™ PowerEdge™ R900 is an excellent server for upgrading critical database applications from Microsoft® SQL Server® 2000 to SQL Server 2008, the latest release of Microsoft's data management platform. This Deployment Guide reviews the approach that Principled Technologies® (PT) and Dell recommend for a tested and validated migration of SQL Server 2000 databases to SQL Server 2008 on the Dell PowerEdge R900 server and Microsoft Windows Server® 2008 Enterprise.

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.

Scope of this Guide

This Deployment Guide reviews the approach that Principled Technologies and Dell recommend for migrating databases from SQL Server 2000 to SQL Server 2008. We cover installing Windows Server 2008 and SQL Server 2008 and migrating user databases. While we do not address in detail the migration of Reporting Services components, Analysis Services databases, and Data Transformation Services (DTS) packages, where applicable, we provide pointers to additional information on these components.

At the time of this writing, Microsoft supports only upgrading SQL Server 2000 (SP4) and SQL Server 2005 (SP2) databases to SQL Server 2008. Consequently, if you are running any SQL Server 7 databases, you will need to upgrade them to SQL Server 2000 SP4 before you can migrate them to SQL Server 2008.

New features in SQL Server 2008

SQL Server 2008 introduces a wide variety of new features and enhances many others that were present in SQL Server 2005. While there are far too many features to discuss in this Guide, the following list highlights some of the more significant ones:

- **Database mirroring.** Microsoft first introduced this feature in SQL Server 2005 and enhanced it in SQL Server 2008. Database mirroring provides data administrators a low-cost disaster recovery solution.
- **Service Broker.** SQL Server Service Broker provides an integrated infrastructure on which developers can create asynchronous messaging applications.
- **Policy-based management.** SQL Server 2008 lets you automatically enforce and monitor policies for database operations. You can also push these policies to your servers.
- **Resource Governor.** The Resource Governor lets you limit the resource consumption of incoming requests by classifying incoming connections as specific workload types and defining limits for those types. You can also redefine the limits in real time.
- **Backup compression.** SQL Server 2008 can automatically compress database backups, a feature previously available only via the purchase of third-party software products.
- **Performance data collection.** SQL Server 2008 lets you store performance data in a centralized database. It also provides enhanced tools for analyzing and reporting on such performance data.
- **New data types.** The new SQL Server 2008 FILESTREAM data type allows you to store large binary data, such as documents or images, directly in an NTFS file system. Other new data types include support for spatial data and the time date type.
- **“Hot add” hardware.** First introduced in SQL Server 2005, hot add memory helps system administrators decrease downtime when important memory upgrades are necessary. On supported hardware platforms, SQL Server 2008 also lets you add one or more CPUs while the program is running.

New features of Windows Server 2008 Enterprise

Windows Server 2008 Enterprise also introduces many new features. As with SQL Server 2008, there are far too many to include in this Guide. The following list mentions a few of the more significant features that may benefit system administrators:

- **Server core.** Windows Server 2008 Enterprise edition allows administrators to install only the functions necessary for a server's role. For example, the server core installation does not include a GUI or the Explorer shell. The reduced number of components should improve efficiency, reduce

maintenance costs, and improve security by reducing the vectors available to viruses.

- **Windows PowerShell™.** This new command-line shell includes over 130 utilities and an admin-focused scripting language to make system administration easier.
- **Self-healing NTFS file system.** If Windows detects a file system error, it spawns a worker thread that attempts to fix this problem. If the problem is recoverable, the system fixes it without the administrator needing to restart it or shut it down.
- **Network Access Protection (NAP).** NAP allows administrators to define health and security policies for the network and to block non-compliant computers from the network.
- **Windows® Deployment Services (WDS).** This replacement for Remote Installation Services (RIS) helps improve the deployment of image-based Windows operating systems. Notably, WDS allows network-based installation of Windows Vista® and Windows Server 2008 Enterprise on computers with no operating system currently installed.
- **BitLocker™ data encryption.** Windows Server 2008 Enterprise adds support for Microsoft's BitLocker, which lets administrators easily and transparently encrypt whole volumes, even across multiple drives.
- **Built-in virtualization.** Windows Server 2008 Enterprise includes a hypervisor, Hyper-V™, which provides native virtualization. Hyper-V allows virtual devices to access the VMBus directly, via Enlightened I/O. This technology can greatly improve efficiency, but does require that the guest OS support Enlightened I/O.

Pre-migration considerations



This section reviews some of the topics you should consider prior to migrating your databases.

32 bit versus 64 bit

In this Guide, we present installation and migration instructions for both 32-bit and 64-bit environments. With the exception of enabling Address Windowing Extensions (AWE), which we discuss below, the steps for the two environments are identical. For the vast majority of cases, we recommend running a 64-bit environment. Nonetheless, we suggest you take a moment to consider the merits of each environment and possible exceptions to our general recommendation.

When you move to a 64-bit environment, you pay some costs. In particular, registers and many data types that previously took 32 bits now take 64 bits. Because the data takes more space, it offers greater range and precision. The main advantage a 64-bit environment brings to a DBMS, however, is a vastly greater address space, which means you can directly address far more RAM.

In a 32-bit environment, you can address up to 4 GB of RAM. Of this 4 GB, Windows by default reserves 2 GB for the operating system. This means that when all code and data for SQL Server do not fit in 2 GB, you pay the overhead of paging data in and out of memory. This limitation can be severe with SQL Server 2008, where a single field of type VARCHAR(MAX) can be as large as 2 GB. While Windows Server does let you alter the default memory allocation with the bcdedit utility, which lets you give up to 3GB to the application processes, the fundamental 4GB limit remains.

To work around this limit, 32-bit Windows Server 2008 provides Address Windowing Extensions, which let an application use memory beyond the 4GB limit. This technique does not really increase the address space, however; the extended memory has restrictions on its use, and you pay a performance penalty for using it. With AWE, the operating system is essentially mapping the extended memory in and out of the virtual address space. While this technique is much faster than paging to disk, it is slower than addressing the memory directly. Also, as we mentioned above, the extra memory has restrictions on its use: SQL Server can use it only to buffer data pages. The extra memory is not available for more connections, thread resources, queues, or other objects.

In contrast, 64-bit Windows Server 2008 supports up to 2 terabytes of physical RAM. The vastly larger address space has the potential to improve performance in almost every area. Obviously, any application that works directly with large amounts of data can benefit from this change. The increased address space also can improve other, less obvious functions, such as running with large numbers of connections and using SQL Server's Analysis Services.

Despite the advantages of the 64-bit environment, under certain circumstances it makes sense to stay in a 32-bit environment. For example, sites running legacy applications or those with other 32-bit dependencies may choose a 32-bit setup. It is also not always feasible to port a mission-critical 32-bit application to 64-bit. For example, there is no 64-bit version of Data Transformation Services (DTS) or SQLMail.

Also, moving to 64-bit does not always improve performance. If you're not experiencing any memory pressure, then, obviously, increasing the amount of available memory will not improve performance. If you're running an application that puts a lot of pressure on the L2 cache, it's actually possible for performance to decline upon moving to 64-bit, because less data will fit in the L2 cache. (Such cache dependencies are rare in large, active databases, and the large L2 cache in the R900's Intel® Xeon® E7320 processors makes it even less likely that this will be a problem.)

For detailed instructions on enabling AWE for 32-bit SQL Server, see [Appendix D](#).

Upgrade paths

The upgrade paths to SQL Server 2008 February CTP that Microsoft supports include SQL Server 2000 SP4 and SQL Server 2005 SP2. A side-by-side migration, as opposed to an in-place migration, avoids most upgrade path issues. For this Guide, we used the backup and restore side-by-side migration method.

For complete details on the upgrade paths Microsoft supports, please see Microsoft SQL Server Books Online at [http://msdn2.microsoft.com/en-us/library/ms143393\(SQL.100\).aspx](http://msdn2.microsoft.com/en-us/library/ms143393(SQL.100).aspx)

Component considerations

In this Guide, we discuss migrating user databases only from SQL Server 2000 to SQL Server 2008. In this section, we touch briefly on other components you might want to migrate and provide references for more information.

- **SQL Server Reporting Services.** In SQL Server 2000, Microsoft's enterprise reporting engine had a tight coupling with Internet Information Services (IIS). SQL Server 2008 removes that dependency. For more information on migration of SQL Server 2000 Reporting Services deployments to SQL Server 2008, see Microsoft SQL Server Books Online at [http://technet.microsoft.com/en-us/library/ms143724\(SQL.100\).aspx](http://technet.microsoft.com/en-us/library/ms143724(SQL.100).aspx)
- **Data Transformation Services.** SQL Server 2005 Integration Services replaced this SQL Server 2000 Extract, Transform, and Load (ETL) component, but Microsoft still supports migration to SQL Server 2008 Integration Services. Microsoft does not offer a 64-bit version of Data Transformation Services (DTS). For more information, see Microsoft SQL Server Books Online at [http://technet.microsoft.com/en-us/library/ms143501\(SQL.100\).aspx](http://technet.microsoft.com/en-us/library/ms143501(SQL.100).aspx)
- **SQL Server Analysis Services.** Microsoft has greatly enhanced Analysis Services since SQL Server 2000. For more information on migrating to SQL Server 2008 Analysis Services, see SQL Server Microsoft SQL Server Books Online at [http://technet.microsoft.com/en-us/library/ms143479\(SQL.100\).aspx](http://technet.microsoft.com/en-us/library/ms143479(SQL.100).aspx)
- **SQL Server Notification Services.** In SQL Server 2008, Microsoft drops this component of SQL Server.

SQL Server 2000 features not present in SQL Server 2008

While Microsoft added many new features between SQL Server 2000 and SQL Server 2008, it also discontinued support for a significant number of commands, syntaxes, and features. If your SQL Server 2000 database depends on these discontinued features, migration to SQL Server 2008 might cause unusual behavior or prevent applications from functioning. Running the SQL Server 2008 Upgrade Advisor should help you locate most of these legacy features, but you should research your particular database environment thoroughly to ensure no migration issues occur. Because your migration scenario spans two release cycles, you must include both (1) all features that changed between SQL Server 2000 and SQL Server 2005 and (2) all features that changed between SQL Server 2005 and SQL Server 2008. Full details on discontinued features are available at the following links:

Discontinued features from SQL Server 2000 to SQL Server 2005: [http://msdn2.microsoft.com/en-us/library/ms144262\(SQL.90\).aspx](http://msdn2.microsoft.com/en-us/library/ms144262(SQL.90).aspx)

Discontinued features from SQL Server 2005 to SQL Server 2008: [http://msdn2.microsoft.com/en-us/library/ms144262\(SQL.100\).aspx](http://msdn2.microsoft.com/en-us/library/ms144262(SQL.100).aspx)

Logins and dependencies: Pre-migration

Windows logins vs. SQL Server logins

When you migrate databases, you must also migrate users' security credentials. While we recommend using Windows Authentication in an Active Directory® environment, we recognize that many legacy installations use a mix of Windows Authentication and SQL Server Authentication. You must plan carefully when porting your SQL Server logins and mapping the database users to those logins. See Appendix G for detailed instructions on transferring logins using Windows Authentication. See Appendix H for detailed instructions on transferring logins using SQL Server Authentication and mapping database users to server logins.

Outside dependencies

Systems administrators, database administrators, and application developers often leverage SQL Server instance-level components and server-level operating system components to achieve their computing goals. You must ensure that these system components

outside the context of the migrated database carry forward to the new server. The most common examples of these components include, but are not limited to, the following:

- **Database-specific SQL Server Agent jobs.** When the database moves to new hardware, the relevant SQL Server Agent jobs must move with it. If any jobs on the legacy SQL Server 2000 server reference the migrated database, you must remove the reference to that database at migration time.
- **Database-specific file shares.** If your stored procedures or applications use file shares on the legacy SQL Server 2000 server, you must either recreate the file share layout in the new environment or update the applications.
- **Linked server connections.** Some database applications use the linked server functionality of SQL Server to access outside data, such as that from Active Directory, Oracle, or DB2. If you have any such dependencies, you must recreate the linked server objects in the new environment.
- **SQLMail configuration.** If you are currently using SQLMail, you will need to decide whether to convert to Database Mail as part of the upgrade. SQL Server 2008 does continue to support SQLMail, but Microsoft does not offer a 64-bit version of SQLMail. Microsoft plans to remove this feature in a future version of SQL Server.
- **Other scripts, batch files, or application connection configurations.** Over time, developers and administrators create maintenance scripts, custom applications, connection configurations, and other components unique to an installation. When you migrate your database(s) to a new environment, you must also migrate these scripts and configurations.

Installing and setting up Windows Server 2008 Enterprise and SQL Server 2008 in this environment



In this section, we focus on installing and setting up Windows Server 2008 Enterprise and SQL Server 2008 on the Dell PowerEdge R900 server. We begin by defining our environment. Next, we present an overview of the steps we took to configure the internal and external drives and to install and configure both Windows Server 2008 Enterprise Edition and SQL Server 2008 Enterprise Edition. Except where we have noted, the steps apply to both x86 (32-bit) and x64 (64-bit) setups.

Defining our environment

We used a Windows domain containing a legacy Windows 2003 server, a new Windows 2008 server, and an Active Directory server. Our legacy server was a Dell PowerEdge 2650 running Windows Server® 2003 x86 Enterprise Edition SP2 and SQL Server 2000 SP4. For our new server, we tested two configurations: (1) Windows Server 2008 Enterprise x64 RC1 and SQL Server 2008 x64, and (2) Windows Server 2008 Enterprise x86 RC1 and SQL Server 2008 x86. We connected a Dell PowerVault MD1000 direct-attached storage array to the R900. Our Active Directory controller was a Dell PowerEdge 1950 running Windows Server 2003 R2 Enterprise Edition SP2. We connected all components via a gigabit switch. Figure 1 illustrates our setup.

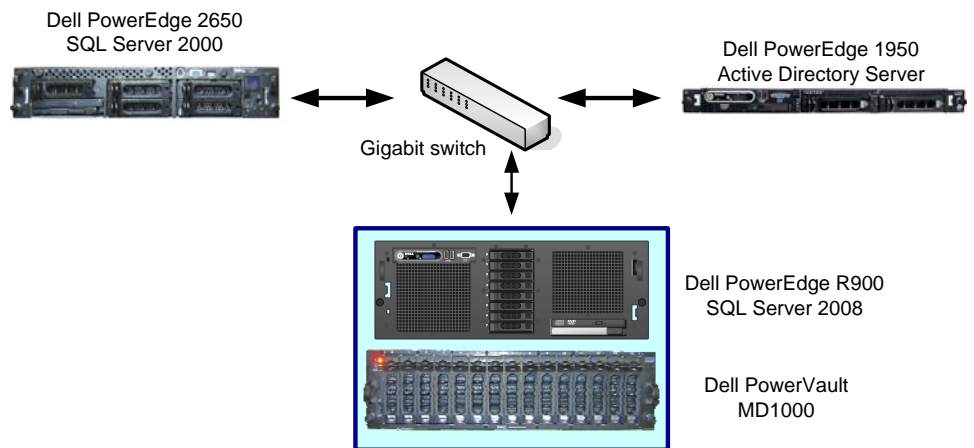


Figure 1. The setup we used in our hands-on testing and research for this Deployment Guide.

Figures 2 and 3 present the hardware and software we used to simulate an Active Directory domain and associated SQL Server 2000 and SQL Server 2008 database servers.

Server	Processor	Memory	Disk
Dell 1950 (Active Directory Server)	2 x Quad Core E5440 Intel® Xeon®	16 GB	2 x Fujitsu® SAS, 15K RPM, 146 GB
Dell 2650 (Legacy Database Server)	2 x Dual Core Intel Xeon (2.8 GHz)	8 GB	5 x Fujitsu Ultra160 SCSI, 36 GB
Dell R900 (migration database server)	4 x Quad Core E7320 Intel Xeon	32 GB	7 x Seagate® SAS, 10K RPM, 73 GB

Figure 2. 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 R2 SP2 x86 Enterprise Edition	N/A
Dell 2650 (Legacy Database Server)	Windows Server 2003 x86 SP2 Enterprise Edition	SQL Server 2000 SP4
Dell R900 (migration database server)	Windows Server 2008 x86 Enterprise Edition RC1	SQL Server 2008 x86 (February CTP)
	Windows Server 2008 x64 Enterprise Edition RC1	SQL Server 2008 x64 (February CTP)

Figure 3. 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 Fujitsu SAS, 15k rpm, 146GB
- **Gigabit Switch**—16 port gigabit



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.

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, Database Mail, 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. Of course, in some circumstances, such as consolidation of a server with few disks, such isolation is not possible.

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 may have very different and even competing I/O characteristics, combining heterogeneous workloads can hurt overall performance.

Below we illustrate the exact drive layout we used (see Figure 4) and briefly describe each disk group. For detailed instructions on how we configured the RAID disk groups, see [Appendix A](#).

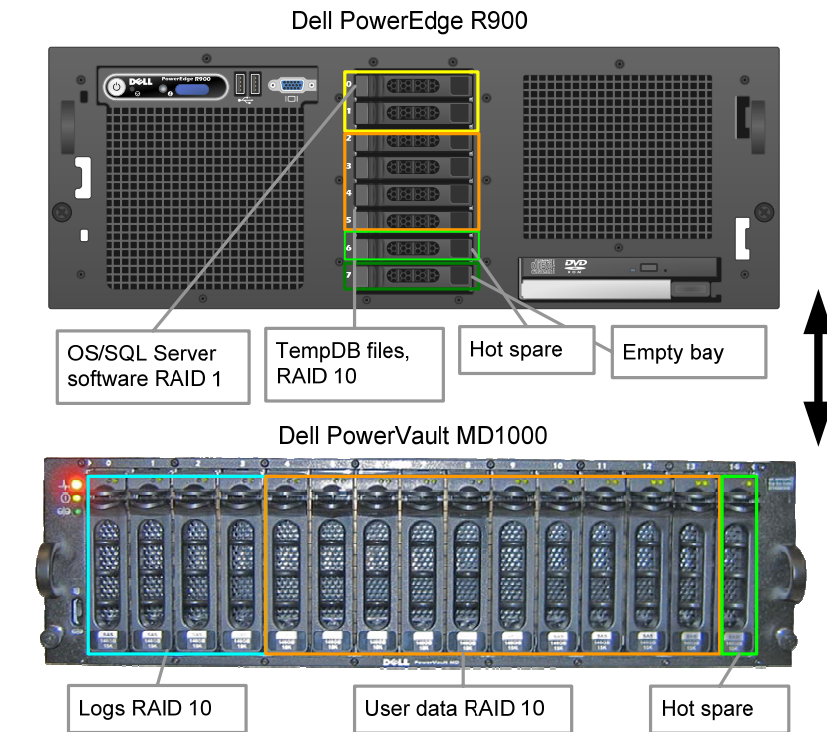


Figure 4. The drive configuration we used in our hands-on testing and research for this Deployment Guide.

Internal disk drives

Seven drives were present in our PowerEdge R900 Server. 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 following four internal drives to the tempdb database, and configured that disk group as RAID 10. We configured the remaining drive as a global hot spare for the other internal drives.

External disk drives

The PowerVault MD1000 contained 15 drives. We split them into two disk groups. We configured the first two drives as RAID 10 for performance and redundancy; this disk group held the SQL Server transaction logs. We configured the next 10 drives at RAID level 10 to hold our SQL Server user-defined databases. Finally, we configured the last disk in the array as a global hot spare, available to both disk groups.

Installing Windows Server 2008

This section provides an overview of the operating system installation process. We include approximate wait times for each step. ([Appendix B](#) provides complete, detailed installation instructions.) Because the Dell PowerEdge Installation and Server Management software did not support Windows Server 2008 Enterprise when we were setting up our server, we installed the operating systems directly from the Microsoft operating system DVDs.



NOTE: Plan on at least 60 minutes for installing Windows Server 2008 Enterprise on the Dell PowerEdge R900 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 DVD into the DVD drive and reboot the system. If a message telling you to press any key to boot from CD or DVD appears, 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 5). (17 minutes, during which Windows resets the server twice)

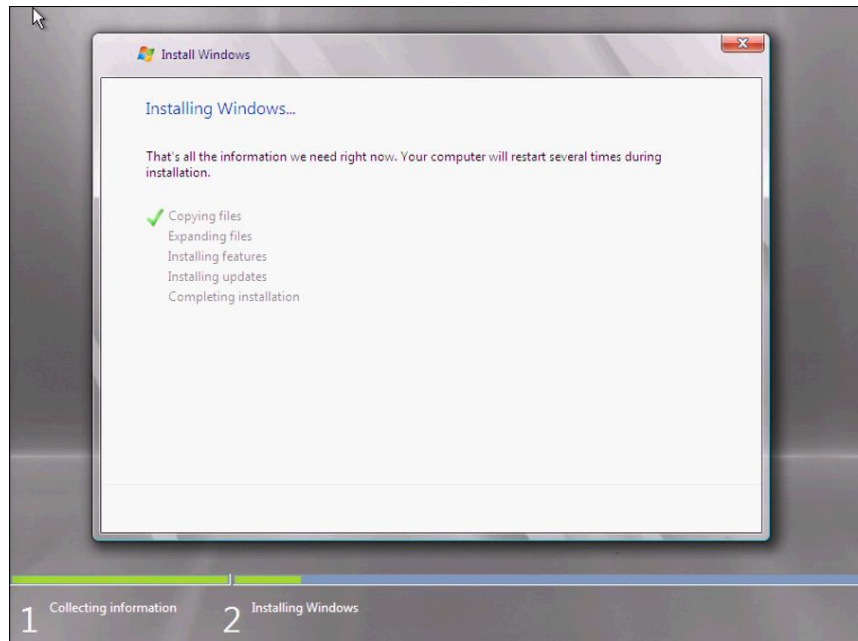


Figure 5. Windows Server 2008 installation progress.

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 6). (3 minutes for us; download times vary)

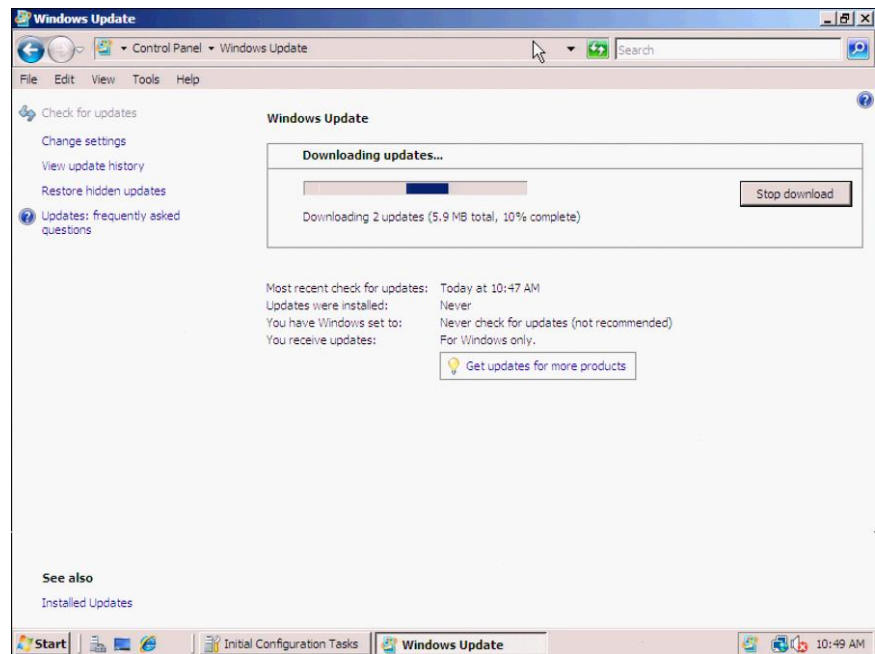


Figure 6. Installing Windows Server 2008 updates.

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 CNAME records to assign aliases to your database servers. Doing so 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 operating system drives in your server using the Disk Management utility that is part of the Windows Server 2008 Administrative Tools. Format as NTFS all SQL Server volumes, including those you use to hold your data,

logs, and tempdb system database. (See Figure 7.) 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).

Virtual disk	Drive location	Raid level	Number of drives	Contents
1	PowerEdge R900	10	4	tempdb files
2	PowerVault MD1000	10	4	Log files
3	PowerVault MD1000	10	10	User data files

Figure 7. Drive configuration and contents.

If you are using an allocation unit size of 64 KB, Windows Server 2008 Enterprise 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 so simplifies 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

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 a member server 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 Microsoft SQL Server Books Online at [http://msdn2.microsoft.com/en-us/library/ms143504\(SQL.100\).aspx#Review_NT_rights](http://msdn2.microsoft.com/en-us/library/ms143504(SQL.100).aspx#Review_NT_rights).

This section provides an overview of the SQL Server 2008 installation process. We include approximate wait times for each step. [Appendix C](#) provides full, detailed installation instructions.



NOTE: We installed SQL Server 2008 using an ISO image we burned to a DVD. Microsoft's download page for the February CTP of SQL Server 2008 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 on the Dell PowerEdge R900 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 DVD into the DVD drive. Click Server components, Tools, Books Online, and Samples; accept the license agreement; and install the prerequisites (See Figure 8). (less than 1 minute)

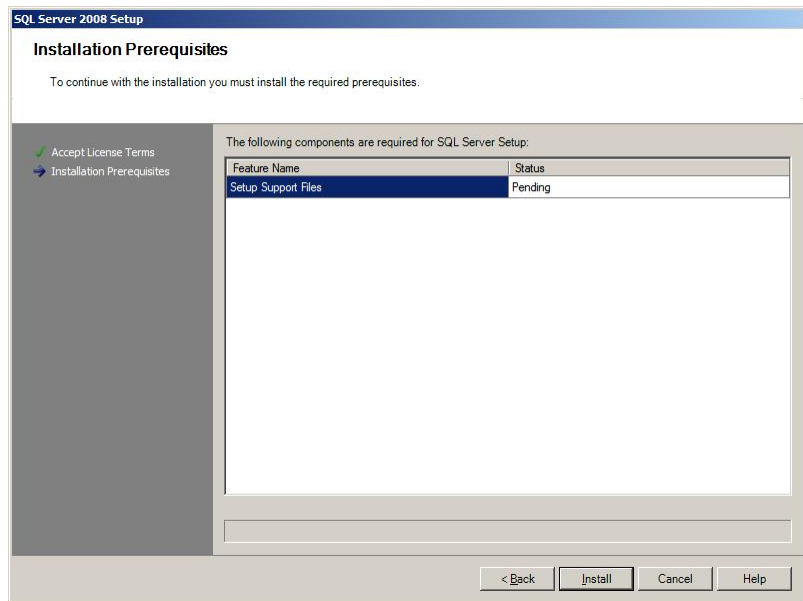


Figure 8. Installing SQL Server 2008 prerequisites.

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, as we did in Figure 9 below. You can install additional components later as necessary.

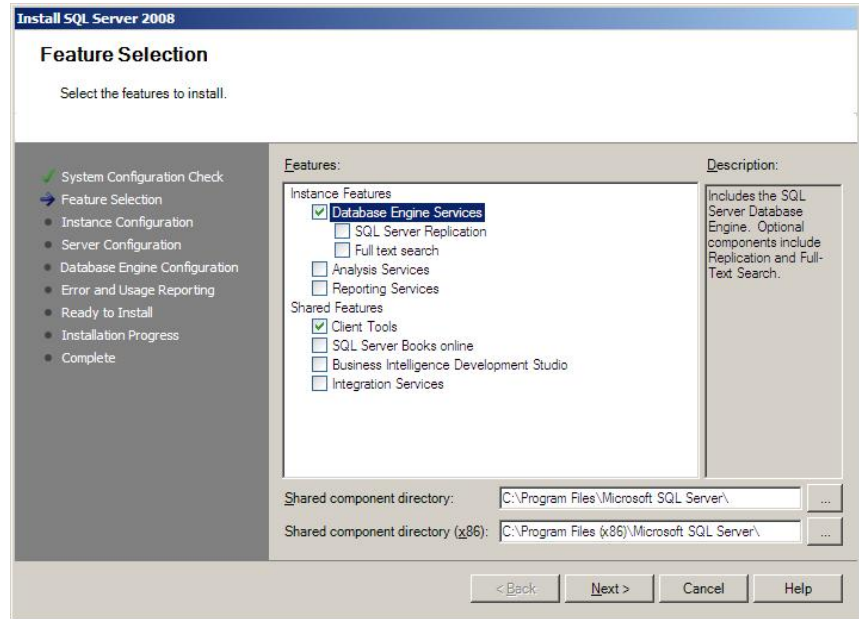


Figure 9. Selecting SQL Server 2008 features.

3. On the following screens, fill in the account information and data file locations (Figure 10). Install. At the confirmation screen, click Next. (7 minutes)

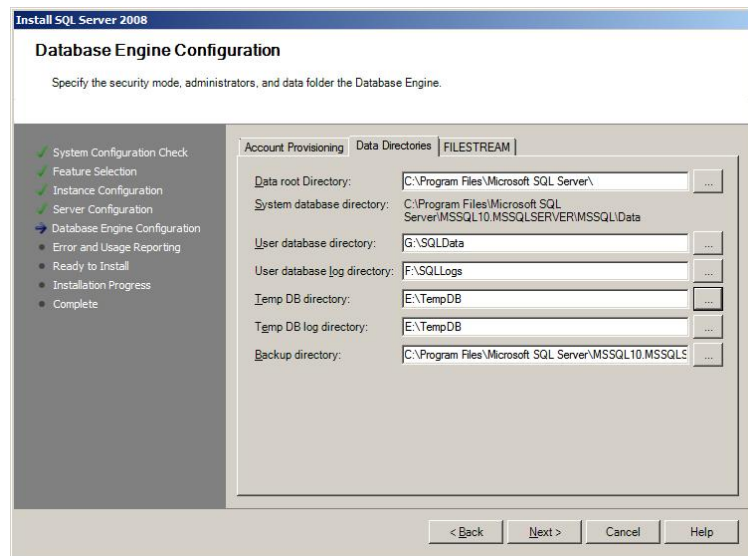


Figure 10. Setting file locations.

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, 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, after the installation process completes you should 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 stops that problem from occurring.

Windows Server 2008 Enterprise 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 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 11).
5. Click Add User or Group, and enter the SQL Server service account name.

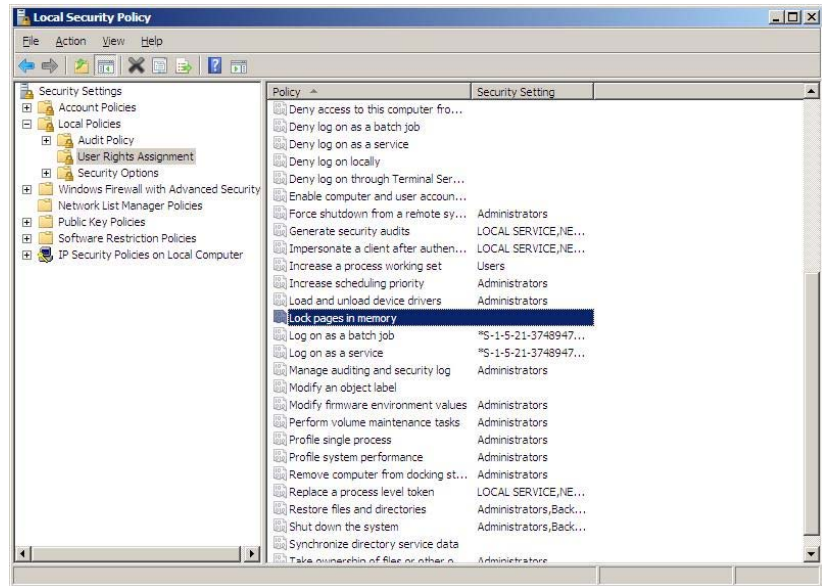


Figure 11. Selecting the Lock pages in memory setting.



BEST PRACTICES: In a 64-bit environment, or a 32-bit environment where SQL Server is not using AWE, leave the Minimum server memory and the Maximum server memory at their defaults of 0 and 2147483647, respectively. Doing so allows SQL Server 2008 to use as much memory as the system makes available. If you must change these settings in a 64-bit environment, make sure that the sum of the maximum memory settings across all processes is less than the amount of physical RAM available. If you are using AWE in a 32 bit environment, however, Microsoft recommends that you set a maximum memory amount. See [Appendix D](#) for details.

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. Microsoft's in-house testing shows a radical performance improvement when using instant data file initialization.

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. 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.



NOTE: If you are configuring a 32-bit environment, you may want to enable AWE to give SQL Server 2008 the ability to access more RAM. See [Appendix D](#) for instructions on enabling AWE.

Migrating to SQL Server 2008



In this section we provide specific details about the migration of user databases from SQL Server 2000 to SQL Server 2008. We begin with a section on analyzing your database with the SQL Server 2008 Upgrade Advisor. We then provide an overview of the processes involved in migrating your database from SQL Server 2000 to SQL Server 2008 on the Dell PowerEdge R900.

Using the SQL Server 2008 Upgrade Advisor to analyze databases

The SQL Server 2008 Upgrade Advisor is a major aid in migration research. This utility, which the SQL Server 2008 setup wizard includes, scans legacy databases and SQL Server components for compatibility issues, features and syntax the newer DBMS does not support, and many other critical components. The utility lets you view reports quickly in the upgrade advisor interface, or you can save reports for later review.

You can install and execute the Upgrade Advisor on machines running Windows XP SP2, Windows Vista, Windows Server 2003 SP1 or later, or Windows Server 2008. The Microsoft .NET 2.0 framework is also a requirement.



BEST PRACTICE: Use the Upgrade Advisor tool on your SQL Server 2000 legacy database and import a trace file to the Upgrade Advisor tool for analysis. The trace file lets the Upgrade Advisor detect issues that might not show up in a simple scan of the database, such as TSQL embedded in applications. Your migration research and planning must account for such instances. You can capture traces of TSQL using SQL Profiler on your SQL Server 2000 server during typical hours and analyze these traces using the Upgrade Advisor.

To install the Upgrade Advisor, use the following steps:

1. Insert the SQL Server 2008 CTP DVD. On the splash screen, click Install SQL Server Upgrade Advisor.
2. Click Next to begin the installation wizard, accept the licensing terms, and click Next.
3. Click Next to accept the default Registration information, click Next to accept the default installation path, and finally click Next to begin the installation.

Once you have installed the SQL Server 2008 Upgrade Advisor, you can use this software to scan your SQL Server 2000 instances for potential migration issues. This section provides a brief walkthrough, [Appendix E](#) gives more detailed instructions.

1. Select Start | All Programs | Microsoft SQL Server 2008 | SQL Server 2008 Upgrade Advisor.
2. Click the Launch Upgrade Advisor Analysis Wizard link, and then click Next to begin.
3. Enter the SQL Server 2000 instance name, and select the features you want the Upgrade Advisor to analyze. Alternatively, click Detect to have the Upgrade Advisor remotely scan the SQL Server 2000 instance and detect which components are running on the SQL Server 2000 instance (See Figure 12).

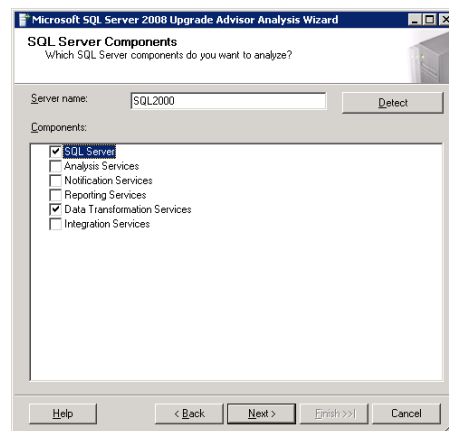


Figure 12. Upgrade Advisor component selection on SQL Server 2000.

4. Provide proper authentication to the legacy SQL Server, and select the database you wish to analyze. If necessary, you can also input SQL script files and trace files here.
5. The packages for Data Transformation Services (DTS) may be present in the file system or in the database itself. If you selected DTS, or if the Upgrade Advisor automatically detected the presence of DTS packages, you must now select the DTS location you want to analyze. Click Run to start the analysis. Run times vary.

Following the analysis of your SQL Server 2000 server, you can view the Upgrade Advisor report, which lists warnings and errors (see Figure 13).

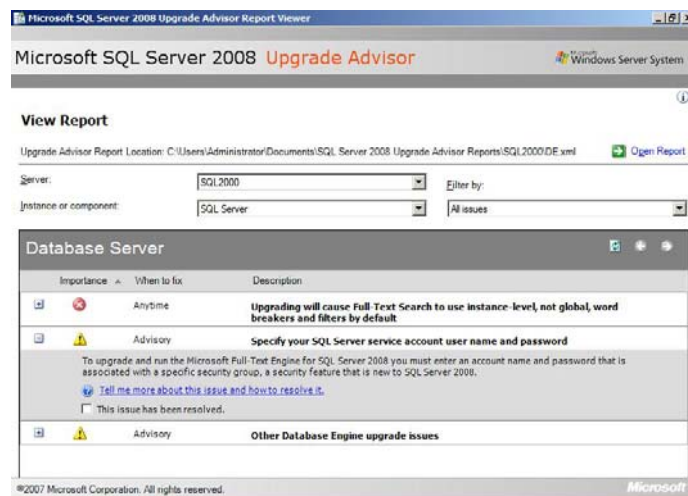


Figure 13. Sample Upgrade Advisor output report.

Implementing side-by-side migration

In this section, we provide an overview of the processes involved in migrating your database from SQL Server 2000 to SQL Server 2008 on the Dell PowerEdge R900. We provide detailed instructions in [Appendix F](#).

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.

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. Creating and verifying our 8 GB backup file on our PowerEdge 2650 took 27 minutes.

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.
2. Perform a full backup of your SQL Server 2000 database, as we did in Figure 14. See [Appendix F](#) for details.

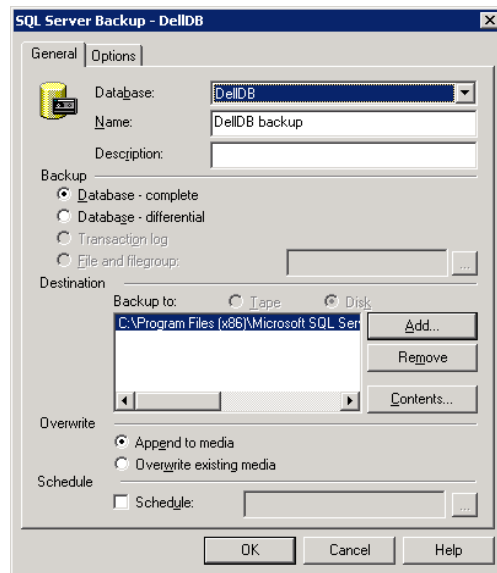


Figure 14. SQL Server 2000 backup configuration.



NOTE: If the time a full backup requires is unacceptable given your migration maintenance schedule, you can take the full backup at a previously scheduled time, and at this point of the migration 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 migration time

significantly. If you go this route, however, be certain to keep your backup chain intact.

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 15). (3 minutes to copy; 2 minutes to restore) See [Appendix F](#) for details.

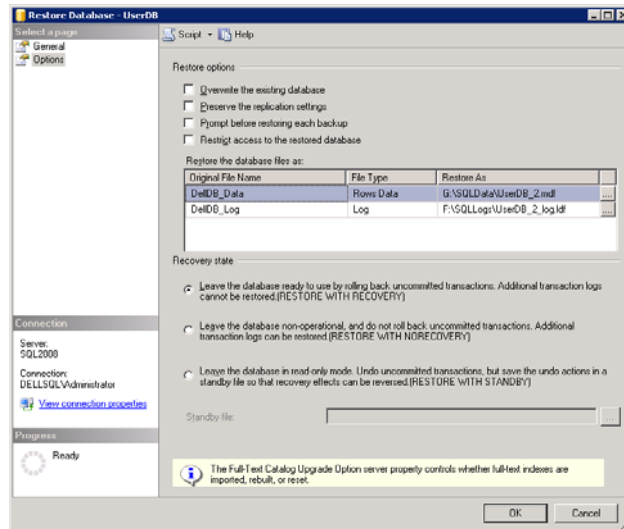


Figure 15. SQL Server 2008 restore configuration.



NOTE: It is important at this step 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 F for details.

Post-migration considerations



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

Compatibility level

SQL Server provides compatibility levels, which determine how certain database features and behaviors work. These levels enable data administrators to upgrade a database to SQL Server 2008 but run that database under a previous version's compatibility mode. After initially restoring your SQL Server 2000 database to SQL Server 2008, the compatibility level will still be 80, which corresponds to SQL Server 2000. To take advantage of all the new features of SQL Server 2008, you must change the compatibility level to 100, which refers to SQL Server 2008. See Appendix F for details.

Logins and dependencies: Post-migration

Windows and SQL Server logins

There are two methods of authenticating to SQL Server 2008: Windows logins and SQL Server logins. You create and administer Windows logins at the Active Directory domain level, 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 migration server is very similar for both Windows authenticated logins and SQL Server authenticated

logins. You should, however, take some extra steps to ensure a smooth migration for SQL Server logins.

Below we describe how to script both Windows authenticated logins and SQL Server authenticated logins to a query window and recreate them on the new SQL Server 2008 server. Step 6 refers solely to the SQL Server login type.

Please 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 both your SQL Server 2000 server and your SQL Server 2008 server. Be sure to have the Object Explorer Details tab open (View | Object Explorer Details).
2. Expand the tree view of the SQL Server 2000 server, browse to the security tab, and click the logins node. In the Object Explorer Details, you will now see a list of all logins on the SQL Server 2000 server.
3. If necessary, use the sorting and filtering options in the Object Explorer Details tab, and take note of which logins you would like to migrate. Select them by clicking; use the standard Windows controls (Ctrl key, Shift key, etc.) to select multiple logins.
4. Right-click the logins you selected, and choose Script Login As | Create To | New Query Window. Be sure to change the connection of this query window to connect to your new SQL Server 2008 server, if it is not doing so already, 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. If you are transferring SQL Server logins, continue with step 6 below.
6. For SQL Server logins, the script you executed in step 5 creates the login, marks it as disabled, and assigns it a random password, but the script does not map the login to the database user. To avoid having a database user that is "orphaned" from a login, use the `sp_change_users_login` stored procedure, which comes with SQL Server, to view orphaned users and to map a

user to a login. [Appendix H](#) provides an example of this process.

SQL Server Agent jobs

Almost all installations schedule SQL Server Agent jobs that run against their databases. You must migrate these jobs to your new SQL Server 2008 server. Locate the jobs you need to migrate in SQL Server Management Studio (on your SQL Server 2000 server) under the SQL Server Agent, right-click them, and choose Script Job As | Create To | New Query Window. Connect to the SQL Server 2008 server, and run the resulting script in a query window.

Other external dependencies

Your pre-migration research may well have yielded a list of items for you to implement now; the list might include file shares, database mail or SQL Mail configurations, stored procedures in system databases, and/or linked servers. Addressing these external dependencies will ensure a smooth finish to your migration.

Summing up

Windows Server 2008 Enterprise and SQL Server 2008 introduce many new features and enhancements. As this Guide has explained, the process of deploying these products on a Dell PowerEdge Server and migrating your SQL Server 2000 databases to the new environment is relatively straightforward; you can perform a basic installation and migration in a couple of hours. Spending some up-front time planning can help you avoid potential problems during your migration.

THIS WHITE PAPER IS FOR INFORMATIONAL PURPOSES ONLY, AND MAY CONTAIN TYPOGRAPHICAL ERRORS AND TECHNICAL INACCURACIES. THE CONTENT IS PROVIDED AS IS, WITHOUT EXPRESS OR IMPLIED WARRANTIES OF ANY KIND. Microsoft, Windows Server, SQL Server, Active Directory, and Windows are registered trademarks and BitLocker, Hyper-V and Windows PowerShell are trademarks of Microsoft Corporation. Intel and Xeon are registered trademarks of Intel Corp. Other trademarks and trade names may be used in this document to refer to either the entities claiming the marks and names or their products. Dell disclaims proprietary interest in the marks and names of others.

For more information, contact Dell.

Information in this document is subject to change without notice.

Appendix A. Methodology: Creating the RAID disk groups

For our installation, we decided to use four RAID groups, each allocated to one of the following:

- Operating system files; SQL Server executable files
- SQL Server tempdb system database
- SQL Server transaction log files
- SQL Server user database files

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

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 6/E showed as Controller 0. The numbering on your system might differ.

Allow at least 10 minutes to complete the installation.

1. 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.
2. We first configure the mirrored RAID for the OS. Select the controller for the internal drives, which is the PERC 6/i. Press Enter.
3. While Controller 1 is highlighted, press F2, and select Create New VD.
4. With the RAID level field highlighted, press Enter to open the drop-down list, and select RAID 1.
5. Using the Tab key for navigation, 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.
6. A message box appears telling you that initializing the RAID is highly recommended, but destructive. Press Enter to clear the message.
7. Highlight Controller 1, press F2, and select Create New VD.
8. With the RAID level field highlighted, press Enter to open the drop-down list, and select RAID 10.
9. Using the Tab key for navigation, move to the Physical Disks section, and use the space bar to select the first four drives (Drives 2, 3, 4, and 5). Note that the drives

you allocated earlier no longer appear in the available list. Then tab to highlight OK, and press Enter.

- 10.** A message box appears telling you that initializing the RAID is highly recommended, but destructive. Press Enter to clear the message.
- 11.** If you have an available disk to use as a hot spare, follow these steps:
 - a. Press <Ctrl><N> to navigate to the Physical Disk Management 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, highlight Yes and click Enter. This disk is now a global hot spare for both Disk Groups on the internal storage.
- 12.** Press <Ctrl><N> to navigate back to the Virtual Disk Management page of the RAID setup utility.
- 13.** Highlight Virtual Disk 0, and press F2.
- 14.** Select Initialization, and then select Fast Init. Highlight OK and press Enter when the system prompts you with any warnings.
- 15.** Highlight Virtual Disk 1, and press F2.
- 16.** Select Initialization, and then select Fast Init. Highlight OK and press Enter when the system prompts you with any warnings.
- 17.** Press F12 to return to the main level of the configuration utility.
- 18.** Select the controller for the drive array (in our case, the PERC 6/E). Press Enter.
- 19.** Highlight controller 0, press F2, and select Create New VD.
- 20.** With the RAID level field highlighted, press Enter to open the drop-down list, and select RAID 10.
- 21.** Using the Tab key for navigation, move to the Physical Disks section. Select the first four drives (Drives 0, 1, 2, and 3) using the space bar. Tab to highlight OK, and press Enter.
- 22.** A message box appears telling you that initializing the RAID is highly recommended, but destructive. Make sure OK is selected, and press Enter.
- 23.** Highlight Controller 0 once again, and press F2. Select Create New VD, and press Enter.
- 24.** With the RAID level field highlighted, press Enter to open the drop-down list, and select RAID 10. Tab to the

Physical Disks section and select the first ten drives (Drives 4 through 13) using the space bar. Note that the drives you allocated earlier no longer appear in the available list. Tab to highlight OK, and press Enter.

- 25.** A message box appears telling you that initializing the RAID is highly recommended, but destructive. Click OK.
- 26.** To use the remaining disk as a hot spare on the PowerVault MD1000:
- 27.** 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.
 - a. 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.
 - b. 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.
- 28.** Exit the RAID utility, and reboot the server.

Appendix B. Methodology: Installing Windows Server 2008 Enterprise

Because the Dell PowerEdge Installation and Server Management software did not support Windows Server 2008 Enterprise when we were preparing this Guide, we installed the operating system directly from the Microsoft DVDs.

Install Microsoft Windows Server 2008 Enterprise 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.

1. Insert the Microsoft Windows Server 2008 Enterprise 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 (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. (17 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. (the 64-bit case took 2 minutes and the 32-bit case took 3 minutes for us; download times vary)
- 15.** Reboot the server. (3 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 using the following process:

1. Click Start.
2. Expand Administrative Tools, and click Computer management.
3. In the left panel, expand Storage, and select Disk Management. If you see the "Initialize Disk" dialog box, 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. Click Next.
7. Leave the drive letter at the default. 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. Windows Server 2008 grays out the option to compress the drive.
 - c. Fill in a volume name if you want the disk to have one.
9. Optionally, select the quick format checkbox.
10. Click Next.
11. On the Completing the New Simple Volume Wizard screen, click Finish.

Appendix C. Methodology: Installing SQL Server 2008



NOTE: We installed SQL Server 2008 using an ISO image we burned to a DVD. Microsoft's download page for the February CTP of SQL Server 2008 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 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.

1. Insert the SQL Server 2008 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. (less than 1 minute for the 32-bit environment, less than 2 minutes in the 64-bit environment)
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. (1 minute for the 32-bit environment, no delay in the 64-bit environment)
8. On the Feature Selection screen, select only those components you need for this particular installation of SQL Server 2008. This practice reduces overhead on your Dell PowerEdge R900. If you plan on running only the database engine, then select just 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. However, if you have to migrate 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. 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. Choose 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..., 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 D. Extending available memory on 32-bit SQL Server 2008 installations – Enabling Address Windowing Extensions (AWE)

Because the R900 supports hardware-enforced Data Execution Prevention (DEP), Windows Server 2008 x86 enabled PAE by default. SQL Server 2008, however, does not use AWE by default. To enable SQL Server 2008 to use AWE to access additional memory, do the following:

1. Ensure the account under which SQL Server is running has the Lock Pages in Memory permission.
 - a. Choose 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..., 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.
2. Enable AWE on your SQL Server 2008 server.
 - a. Log in to SQL Server Management Studio by selecting Start | All Programs | Microsoft SQL Server 2008 | SQL Server Management Studio.
 - b. Enter the necessary credentials for full sysadmin SQL Server access.
 - c. Open a new query window by right-clicking the server in Object Explorer, and selecting New Query.
 - d. Run the following commands to enable AWE:

```
sp_configure 'show advanced options', 1
RECONFIGURE
GO
sp_configure 'awe enabled', 1
RECONFIGURE
GO
```
 - e. Click Execute.
3. Configure Max Server Memory setting.



NOTE: Microsoft strongly recommends that you set a value for max server memory whenever you enable AWE. The operating system will not swap the AWE memory to the page file, and operating system performance can suffer if there is not enough memory available.

- a. Log in to SQL Server Management Studio by selecting Start | All Programs | Microsoft SQL Server 2008 | SQL Server Management Studio.
- b. Enter the necessary credentials for full sysadmin SQL Server access.
- c. Open a new query window by right-clicking the server in Object Explorer, and selecting New Query.
- d. Run the following commands to configure min and max server memory. The following example sets the minimum server memory at 1 GB. Because we had 32 GB available in our PowerEdge R900, we set the max server memory to 24 GB, allowing 8 GB for the operating system and other needs.

```
sp_configure 'min server memory', 1024
RECONFIGURE
GO
sp_configure 'max server memory', 24576
RECONFIGURE
GO
```

4. Restart the SQL Server service by selecting Start | All Programs | Microsoft SQL Server 2008 | Configuration Tools | SQL Server Configuration Manager. Right-click the SQL Server service, and select Restart.
5. Verify that SQL Server is using AWE by looking in the SQL Server error log:
 - a. Open SQL Server Management Studio.
 - b. In Object Explorer, browse to Management | SQL Server Logs, and then double-click the Current log.

If the operating system enabled AWE, you will see the following message upon restart of the SQL Server service:

"Address Windowing Extensions is enabled. This is an informational message only; no user action is required."

Appendix E. Running the SQL Server 2008 Upgrade Advisor

In this example, we walk through the steps to install and run the SQL Server 2008 Upgrade Advisor and save reports. Where applicable, we provide approximate times.

1. Insert the SQL Server 2008 DVD. If the splash screen does not appear, navigate to the \Servers directory and double-click splash.hta. On the splash screen, click Install SQL Server Upgrade Advisor.
2. Click Next to begin the installation wizard, accept the licensing terms, and click Next.
3. Click Next to accept the default Registration information, click Next to accept the default installation path, and finally click Next to begin the installation. (less than 1 minute)
4. At the Completing the Microsoft SQL Server 2008 Upgrade Advisor screen, click Finish to exit the setup.
5. Start the Upgrade Advisor by selecting Start | All Programs | Microsoft SQL Server 2008 | SQL Server 2008 Upgrade Advisor.
6. Click the Launch Upgrade Advisor Analysis Wizard link.
7. Click Next on the Welcome screen.
8. On the SQL Server Components screen, by default the Upgrade Advisor populates the server name field with the local computer name. If you need to scan a remote server, type the computer name or IP address of the SQL Server you want to analyze and do one of the following:
 - a. Click Detect to allow the Upgrade Advisor to scan the components on the SQL Server instance you specified. (We chose this option.)
 - b. Select the components of the SQL Server instance you would like the Upgrade Advisor to scan.



NOTE: If you are analyzing an instance of SQL Server Reporting Services, you must install and run the SQL Server 2008 Upgrade Advisor software on the hardware where SQL Server 2000 Reporting Services is running, because the Upgrade Advisor cannot scan Reporting Services resources across the network.

Do not enter the SQL Server instance name. The Upgrade Advisor will scan the server you specify and check for multiple instances of SQL Server.

If you are scanning clustered components of SQL Server on a failover cluster SQL Server instance, enter the failover cluster instance name. If you are scanning non-clustered components, such as the workstation tools, on a failover cluster, enter the computer name of the specified node.

9. On the Connection Parameters screen, if necessary specify the Authentication Type and credentials.
10. On the SQL Server parameters screen, select one or more databases and, if you so desire, supply a trace file or SQL script file to analyze. Should you wish to use a trace file, you must generate it before you begin the Upgrade Advisor. For instructions about how to generate a trace file, see [http://msdn2.microsoft.com/en-us/library/ms187929\(SQL.100\).aspx](http://msdn2.microsoft.com/en-us/library/ms187929(SQL.100).aspx)
11. If you specified analysis of Data Transformation Packages (or if the detection process discovered DTS on the SQL Server 2000 server), you should now choose whether you want the Upgrade Advisor to scan for (a) DTS packages on the SQL Server 2000 server or (b) DTS package files stored on the file system. We kept the default of Analyze DTS packages on Server. Click Next to continue.
12. On the Confirm Upgrade Advisor Settings screen, review your choices, and then click Run to begin the analysis. (3 minutes)
13. When the analysis completes, click Launch Report to view the analysis report.
14. You can now use the drop-down filter tools to view report items by component or by issue severity.



NOTE: If you want to export the report to comma separated value (CSV) format, click Export Report in the lower right corner of the Upgrade Advisor interface.

Appendix F. SQL Server 2000 to SQL Server 2008 side-by-side migration: Step by step

In this appendix, we give detailed instructions about a basic side-by-side migration of a user database from a SQL Server 2000 server to a SQL Server 2008 server. There are multiple methods of accomplishing this task, including using TSQL commands and automated tools. Here we discuss performing 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 rights on SQL Server 2000.
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.

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
```

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

10. Keep the default of complete backup.
11. 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.
12. Click OK to close the Backup Device Location window, and then click OK to close the Select Backup Destination window.



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.

13. 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.

14. Click OK to begin the SQL Server 2000 database backup. (27 minutes; backup times vary with database size and backup device configuration)
15. After the backup completes, in Enterprise Manager, under Databases, right-click the database, and choose All Tasks | Take Offline. The old database is now offline, so SQL Server 2000 will not allow any connections to it.
16. Log out of Windows on the SQL Server 2000 server.
17. Log in to Windows on the SQL Server 2008 server.
18. In Windows Explorer, navigate to either the shared network location or the portable storage device that you used to store your database backup.
19. Copy the backup file, and paste it to a local directory of your choice. (for us, 3 minutes over a 1 Gb network connection to a shared folder to copy the 8 GB backup; copy time varies with network speed and file size)
20. Open SQL Server Management Studio by selecting Start | All Programs | Microsoft SQL Server 2008 | SQL Server Management Studio.
21. Select Database Engine for Server type, and type the server name in Server name. For Authentication, we used Windows Authentication.
22. In the Object Explorer pane, right-click Databases, and select Restore Database.

- 23.** Enter the name of the database in the “To database” box.



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.

- 24.** Click the “From device” radio button, and then click “...” to browse to the local folder where you copied the backup file.
- 25.** 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.
- 26.** 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.
- 27.** Click OK to begin the restore. Note the progress indicator in the lower left of the Restore window. (2 minutes for us; restore time varies with database size and server and disk subsystem speed)
- 28.** 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:
- Right-click the database in Object Explorer, and select Properties.
 - Click Options on the left side to access database options.
 - Change Compatibility level to SQL Server 2008 (100).
 - Scroll down to Restrict Access, and change to MULTI_USER.
 - Click OK.
 - Click Yes to agree to shutting down other connections.

Appendix G. Transferring Windows logins via SQL Server Management Studio

In this appendix, we provide detailed instructions on migrating transferring Windows-based logins from a SQL Server 2000 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 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 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. Windows Logins will appear as Login Type 0. 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 (see Figure 16).

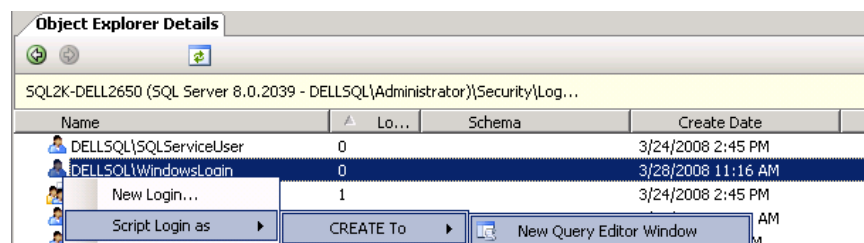


Figure 16. Scripting logins on SQL Server 2008.

8. On the Connect to Database Engine screen, change the Server name to your SQL Server 2008 server. 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 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 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. Find the unmapped database users.

- a. 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.
- b. 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.
- c. Right-click the relevant database, and select New Query.
- d. Enter the following command, and click Execute:

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

2. For each user output from the above step, map these "orphaned" users to the appropriate matching SQL Server login by taking the following steps.
 - a. Open a query window on the SQL Server 2008 using steps a to c above.
 - b. For each login you wish to map, enter the following command, and click Execute:

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

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."

3. 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
```

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.

Principled Technologies is a registered trademark of Principled Technologies, Inc.
All other product names are the trademarks of their respective owners

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 # 89997614](#)