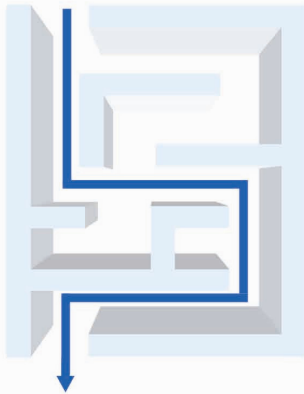


DEPLOYMENT



GUIDE

SQL SERVER® 2005 x64

ON

WINDOWS SERVER® 2008 ENTERPRISE x64

ON

DELL™ POWEREDGE™ 2970

Table of contents

Introduction	3
New features of Windows Server 2008 Enterprise.....	3
Installing and setting up Windows Server 2008 Enterprise x64 and SQL Server 2005 x64 in this environment	4
Defining our environment	4
Configuring the RAID disk groups.....	5
Installing Windows Server 2008 Enterprise x64.....	8
Configuring the operating system drives	10
Installing SQL Server 2005	11
SQL Server 2005 post-installation tasks.....	15
Next steps: Best practices for administering and monitoring SQL Server 2005	20
Administering SQL Server 2005 in this environment	20
<i>AUTOGROWTH</i>	21
<i>AUTO CLOSE</i>	21
<i>One tempdb data file per core</i>	22
<i>Processor and memory parameters</i>	22
Monitoring SQL Server 2005 in this environment.....	23
<i>Counters available from the Reliability and Performance Monitor</i>	23
<i>Dynamic management views and functions</i>	24
Summing up	26
Appendix A. Methodology: Creating the RAID disk groups.	28
Appendix B. Methodology: Installing Windows Server 2008 Enterprise.....	31
<i>Setting IP address, subnet mask, and domain information</i>	32
<i>Configuring data drives</i>	33
Appendix C. Methodology: Installing SQL Server 2005	34
<i>Using the ALTER DATABASE command to modify the location of user database or tempdb files and logs after an installation</i>	37
About Principled Technologies.....	40

Introduction



Dell PowerEdge 2970

The Dell™ PowerEdge™ 2970 is an excellent server for deploying the enterprise database management platform Microsoft® SQL Server® 2005. This Deployment Guide reviews the approach that Principled Technologies® (PT) and Dell recommend for implementing a tested and validated solution for SQL Server 2005 on the Dell PowerEdge 2970 9th generation server and Microsoft Windows Server® 2008 Enterprise x64.

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 the many features in SQL Server 2005.

New features of Windows Server 2008 Enterprise

Windows Server 2008 Enterprise x64 introduces many new features. The following list mentions a few of the more significant enhancements that can benefit database and systems 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 lower number of mandatory 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 will fix it without requiring the administrator to restart or shut down.
- **Network access protection (NAP).** NAP lets administrators define network health and security policies and to block non-compliant computers from the network.
- **Windows Deployment Services (WDS).** This replacement for Remote Installation Services (RIS) improves 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 present.

- **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 it does require that the guest OS support Enlightened I/O.

Installing and setting up Windows Server 2008 Enterprise x64 and SQL Server 2005 x64 in this environment

In this section, we focus on installing and setting up Windows Server 2008 Enterprise x64 and SQL Server 2005 x64 on the Dell PowerEdge 2970 server. We begin by defining the environment we used for our sample setup and then discuss how to configure the server's drives.

Defining our environment

The operating system on our server was Windows Server 2008 Enterprise x64, Service Pack 1. The version of SQL Server 2005 we used was Build 9.00.3054, dated March 23, 2007. The build number reflects Service Pack 2, the latest available service pack release on SQL Server 2005 as of the start of our work on this Guide.

Figure 1 presents the setup we used in our hands-on testing and research for this Guide. The heart of the setup was a Dell PowerEdge 2970 with the following components:

- two Dual-Core AMD Opteron™ 2222SE processors
- 32 GB of RAM
- three 73GB Fujitsu® 10,000 RPM SAS disks
- Dell PERC 5/i integrated RAID controller and PERC 5/E RAID controller
- two Broadcom® BCM5708C NetXtreme II™ GigE integrated NICs

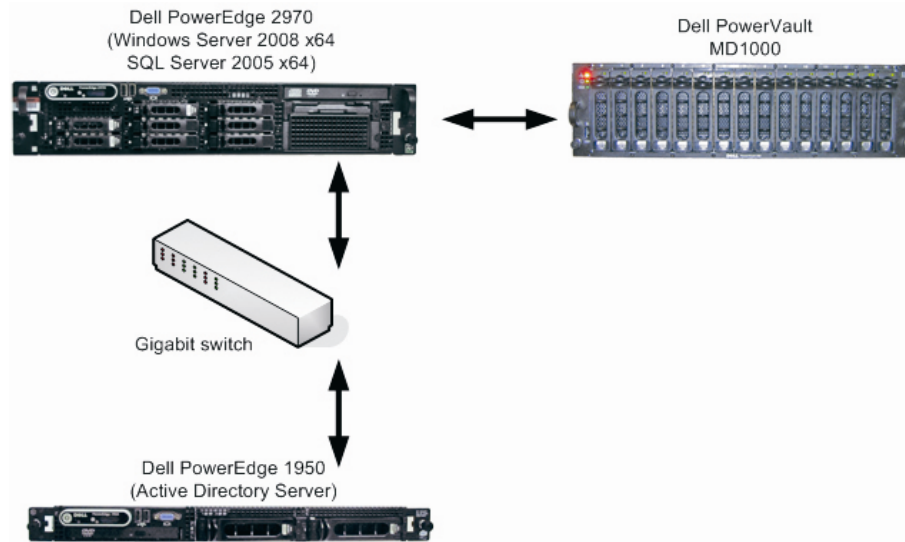


Figure 1. The setup we used in our hands-on testing and research for this Deployment Guide.

We connected the Dell PowerEdge 2970 server to the Dell PowerVault™ MD1000 external disk enclosure. The PowerVault MD1000 contained 15 146 GB Fujitsu 15,000 RPM SAS disk drives. We also connected the PowerEdge 2970 server to an Active Directory® domain controller via a gigabit switch. Our Active Directory server was a Dell PowerEdge 1950, running Windows Server 2003 x64 Enterprise Edition.



BEST PRACTICES: 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>

Deploy SQL Server 2005 in an Active Directory domain and use Windows authentication. Doing so allows you to take advantage of the integrated security and centralized management features of Active Directory

Configuring the RAID disk groups

The first step after setting up your hardware is to configure your drive layout in both the PowerEdge 2970 and the direct-attached PowerVault MD1000. In accordance with Microsoft’s best practices, we allocated our disks so we had separate tempdb database files, user databases, and transaction logs.

Tempdb database

SQL Server 2005 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, aggregate building 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 PRACTICES: Separate the tempdb, transaction log files, and user database files onto their own disks or, when possible, separate disk groups. Doing so can help yield better I/O performance by allowing the server to access data in all of them at the same time. Of course, in some circumstances, such as consolidation or a server with few disks, such isolation is not possible.

If your system uses the full-text search features of SQL Server 2005, 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 (Figure 2) and briefly describe each disk group. For detailed instructions on how we configured the RAID disk groups, see Appendix A.

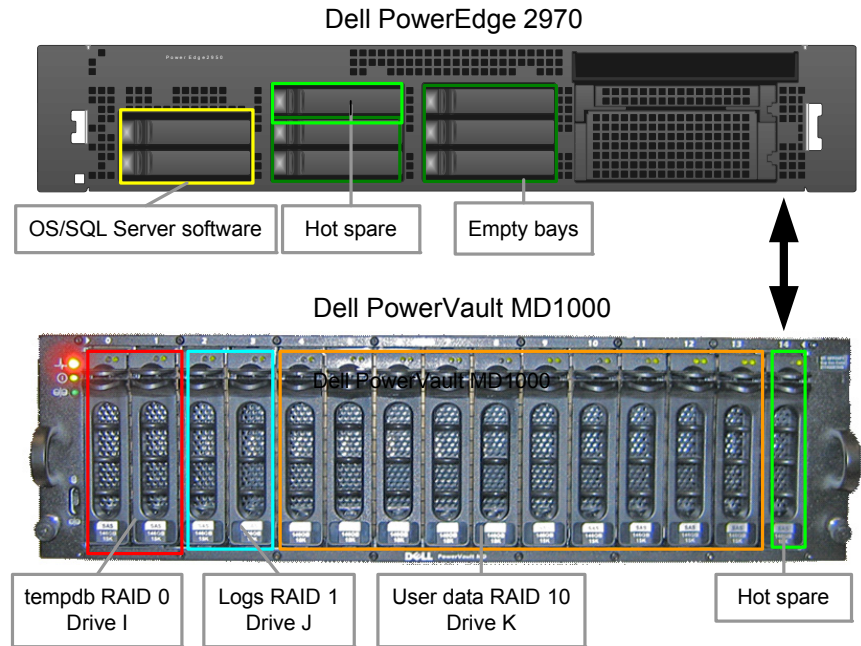


Figure 2. The drive configuration we used in our hands-on testing and research for this Deployment Guide.

Internal disk drives

We had three disk drives internal to the PowerEdge 2970. We mirrored two of the internal PowerEdge drives for the operating system software and SQL Server 2005 software. We configured the remaining disk drive as a hot spare.

External disk drives

The PowerVault MD1000 contained 15 disk drives. We split those drives into three groups, each using a different type of RAID. We configured the first two drives as RAID 0 for optimal performance and allocated tempdb to them. Because SQL Server rebuilds the tempdb database upon every restart of the SQL Server service, we opted for the performance of RAID 0 over the redundancy of RAID 1. We configured the next two drives as RAID level 1 to hold the SQL Server transaction logs. Redundancy is key here, because we do not want to lose transaction logs. We configured the next 10 drives as RAID level 10 for both performance and redundancy. This disk group held our SQL Server user-defined databases. Finally, we configured the last disk in the array as a hot spare.

Installing Windows Server 2008 Enterprise x64

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 x64 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 x64 on the Dell PowerEdge 2970 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—10 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 Windows is loading files message appears, followed by a Windows booting loading bar. (4 minutes)
- 2.** Set the basic location information, enter your product key if necessary, select full install, and enter the information the installation software requires. Start the installation (see Figure 3). (20 minutes, during which Windows resets the server twice)

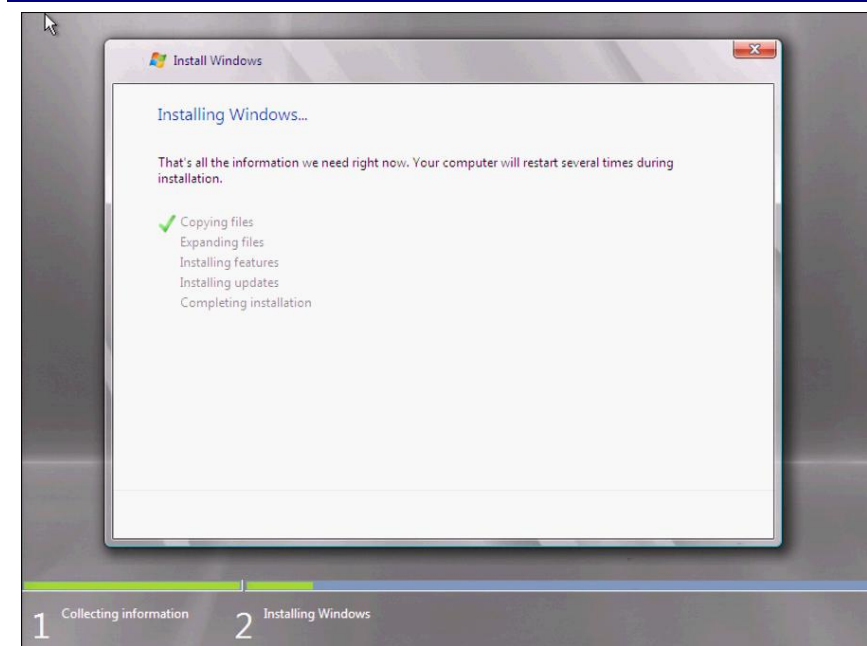


Figure 3. Windows Server 2008 installation progress.

3. Set your password; enter configuration information, including the IP addresses and domain information; and reboot the system. (2 minutes, 30 seconds)
4. Download and install updates (see Figure 4). (2 minutes for us; download times vary)

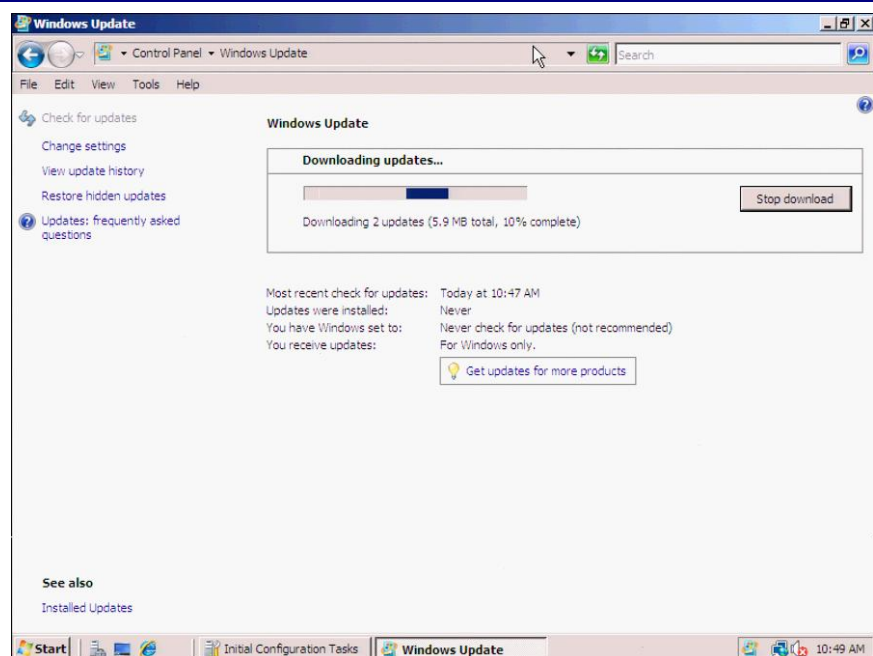


Figure 4. Installing Windows Server 2008 updates.

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 2005 resources will 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 5 for a tabular presentation of our disk layout.) 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, because 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
0	Server	1	2	OS/SQL binaries
1	PowerVault	0	2	tempdb files
2	PowerVault	1	2	Log files
3	PowerVault	10	10	User data files

Figure 5. Drive configuration and contents description

Do not use compressed drives. Windows Server 2008 Enterprise x64 does not offer the option to compress drives when the allocation unit size is 64 KB. Moreover, SQL Server 2005 can only use compressed volumes for read-only secondary data files. In some cases, it may be beneficial to create multiple files and filegroups for your user databases. Doing so will simplify

management and in special cases, benefit performance. In the case of the tempdb database, use one data file per core.



NOTE: After SQL Server 2005 installation is complete, you can add or modify data files and filegroups in the SQL Server Management Studio by right-clicking the database in question, selecting Properties, and then clicking Files or Filegroups. An alternative method to modifying files and filegroups is to use the Transact SQL command ALTER DATABASE. We discuss this topic further in Appendix C.

Installing SQL Server 2005

As we explained above, we ran our Active Directory on a separate server. We highly recommend against deploying SQL Server on any Active Directory controller. Active Directory processes will 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 will create the local groups it needs for its security purposes. For more information on the Windows Domain accounts you need for SQL Server service accounts, please see Microsoft SQL Server Books Online at <http://msdn2.microsoft.com/en-us/library/ms143504.aspx>

This section provides an overview of the SQL Server 2005 installation process. We include approximate wait times for each step. Appendix C provides complete, detailed installation instructions. Appendix C also includes instructions for creating a user database and for using the ALTER DATABASE command to move an existing database.



NOTE: Plan on at least 30 minutes for installing SQL Server 2005 on the Dell PowerEdge 2970 server. Each step below takes at least 1 minute. We provide estimated times for each step in parentheses. Those times exclude data entry time.

1. Insert the SQL Server 2005 DVD into the DVD drive. Click Server components, Tools, Books Online, and Samples;

accept the license agreement; and install the prerequisites (see Figure 6). If the DVD drive does not autorun, you can reach the appropriate screen by double-clicking the splash.hta file, located in the \Servers directory under the appropriate Edition folder. (1 minute)

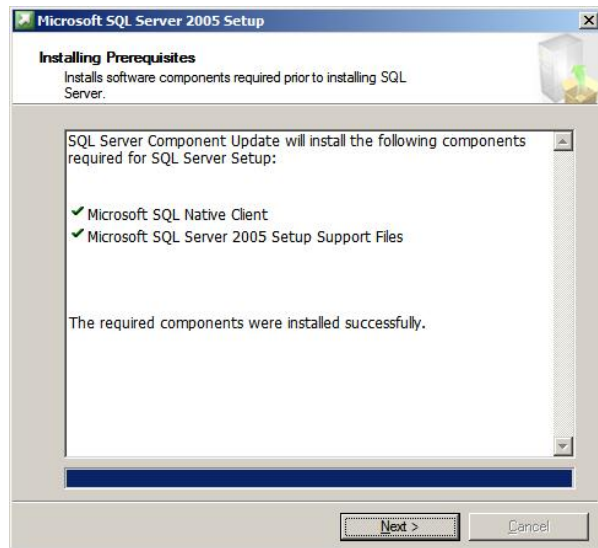


Figure 6. Installing SQL Server 2005 prerequisites.

2. Click Next to proceed to the Microsoft SQL Server Installation 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. You can install additional components later as necessary. (See Figure 7.)



NOTE: Unless absolutely necessary, we recommend not installing or utilizing SQL Server Notification Services in SQL Server 2005. Microsoft drops this component of SQL Server in SQL Server 2008, the next version of SQL Server, and so developers may have to redesign and rewrite applications that depend on SQL Server Notification Services.

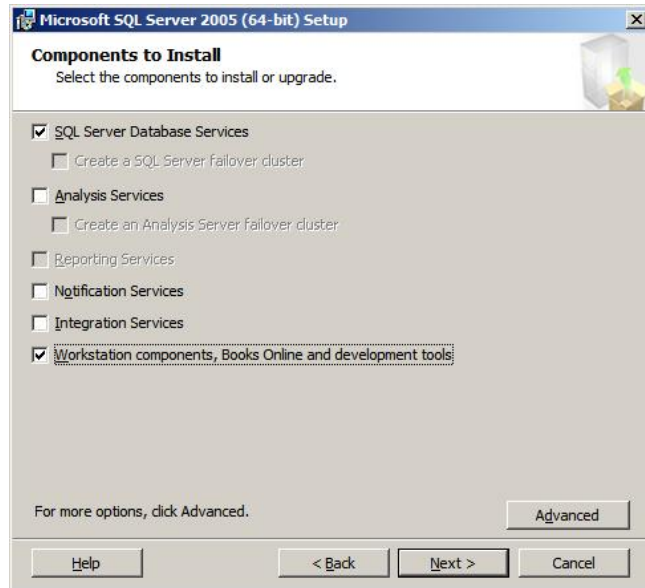


Figure 7. Selecting SQL Server 2005 features.

3. On the following screens, select Default or Named Instance, fill in the account information, select collation settings, and finally review your installation configuration. Install. (See Figure 8.) At the confirmation screen, click Next. (8 minutes)

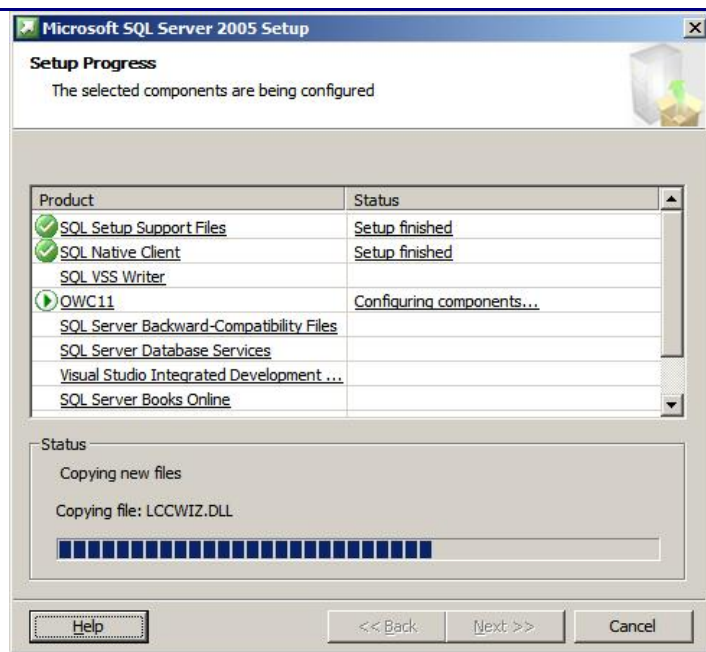


Figure 8. Installation of Microsoft SQL Server 2005.



NOTE: As a part of the installation, the wizard installs the Visual Studio IDE. If you receive a warning that Visual Studio 2005® and Windows Server/Windows Vista have known compatibility issues, click Run Program to continue the installation. At the end of the base installation you will install service packs and patches; according to Microsoft, these will resolve compatibility issues.

4. The Completing Microsoft SQL Server 2005 Setup screen includes a hyperlink to the Summary log file, which can be helpful in troubleshooting failed installations. Should you need it, this log file's default location is C:\Program Files\Microsoft SQL Server\90\Setup Bootstrap\LOG. Click Finish to complete the install.
5. Download and install the latest Service Pack and cumulative hotfixes for SQL Server 2005 from the Microsoft Downloads Center (<http://www.microsoft.com/downloads>). (See Figure 9.) At the time of this writing, SP2 was the latest service pack release for SQL Server 2005. Reboot if prompted to do so. (download time is about 35 minutes; installation time is 4 minutes)

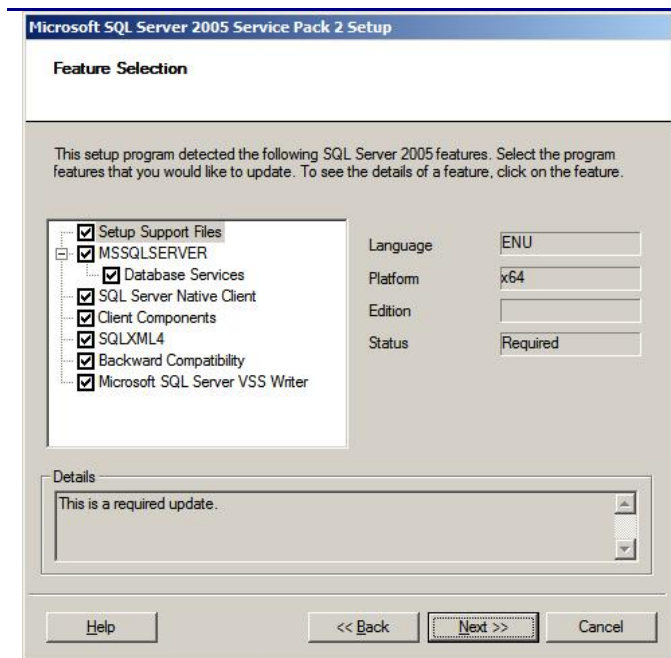


Figure 9. Selecting components of Service Pack 2.



NOTE: In Windows Server 2008, the SQL Server 2005 installation software does not automatically add local administrators to the sysadmin SQL Server 2005 fixed server role. At the conclusion of the installation of SP2, the installation software will provide an option to launch the User Provisioning Tool. This tool lets you allow administrator users full access to SQL Server 2005. Before running the User Provisioning Tool, you may need to reboot your server to completely install all necessary patches and start all SQL services. Otherwise, you may receive an error from the provisioning tool because the SQL Service may not be running as a result of the SP2 install. If you want to run the provisioning tool manually after rebooting, you can find a link to the user provisioning tool within the SQL Server Surface Area Configuration Tool; look for Add New Administrator.

SQL Server 2005 post-installation tasks

As an administrator, after installing SQL Server 2005, you probably want to optimize your new installation before you put it into production. In this section, we discuss best practices for memory settings, file initialization, and placing your SQL Server data, log, and tempdb files in appropriate volumes.

Retaining data in memory

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 releases memory to relieve memory pressure and swapping. Another process, however, can always request a substantial amount of memory and cause the OS to swap some of SQL Server's memory to disk before SQL Server has the chance to react.

Windows Server 2008 Enterprise provides a setting that can prevent this from happening by letting SQL Server 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 2005 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 10.)
5. Click Add User or Group, and enter the SQL Server service account name.

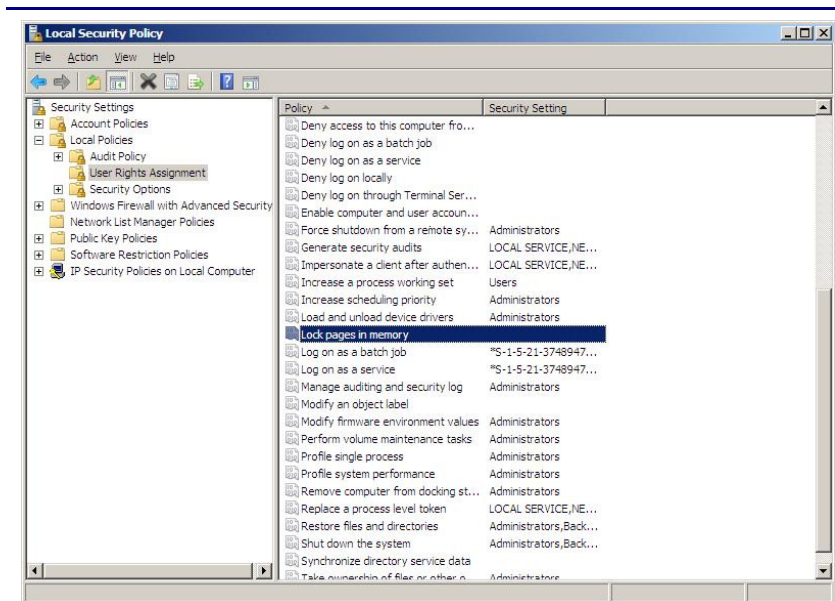


Figure 10. Selecting the Lock pages in memory setting.



BEST PRACTICE: When possible, leave the Minimum server memory and the Maximum server memory at their defaults of 0 and 2147483647, respectively. This practice allows SQL Server 2005 to use as much memory as the system makes available. If you must change these settings, make 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. (See the discussion of AUTOGROWTH below.) 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 2005 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 SQL Server service account name.

Changing tempdb and default user database file locations

The default installation of SQL Server 2005 places system databases, user databases, and all transaction logs under the SQL Server installation directory. To take advantage of your drive configuration, and to follow Microsoft’s best practices, we recommend modifying server settings to reflect the different areas where the user data files, transaction log files, and tempdb files will reside.



NOTE: The SQL Server installation routine creates certain local groups and Access Control Lists (ACL) to enable the SQL Service Account to function properly. When modifying where you store SQL Server files, pay careful attention to permissions and what file access users need. When setting up new folders for data storage, grant full control to the SQL Server service account on these folders. For specific information on permissions and ACL considerations for SQL Server service accounts, see Microsoft SQL Server Books Online at http://msdn2.microsoft.com/en-us/library/ms143504.aspx#Reviewing_ACLs

1. Add the SQL Service account to the local users group: select Start | Administrative tools | Computer Management | Local Users and Groups | Groups, and double-click the Users group. Add the SQL Service domain user service account.

2. Create a folder on the root of each new virtual disk you configured during system setup. In our environment we had the following drive configuration:

OS Folder	Contents
I:\Tempdb\	TempDB
J:\SQLLogs\	Transaction logs
K:\SQLData\	User data

3. Right-click each of the above folders, select Properties, and then select the Security tab.
4. Click Edit, then Add, and then add the SQL Service domain account to the ACL for this folder. Click the Full Control checkbox to allow SQL Server the maximum permission on this folder. (See Figure 11.)

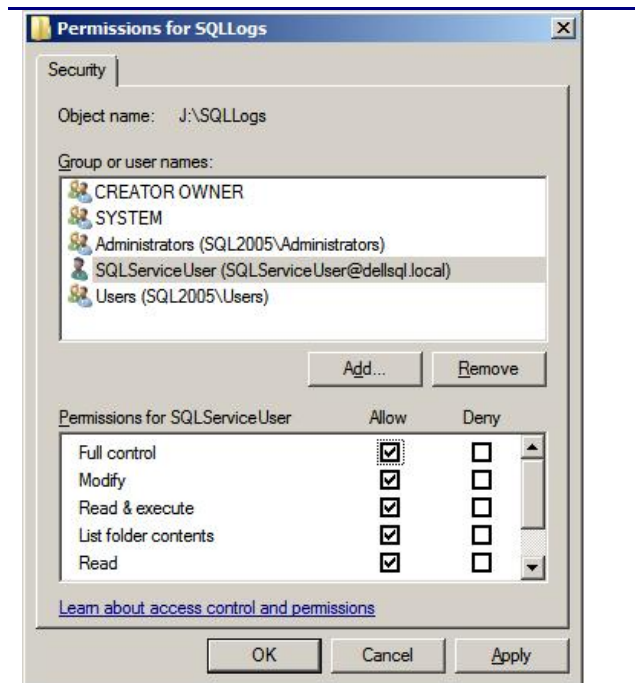


Figure 11. Permissions for SQLLogs.

5. Open SQL Server Management Studio, and connect to the new server using Windows Authentication. Right-click the server in Object Explorer, and select Properties. Click Database Settings in the left pane and modify the default database locations to reflect your drive configurations. (See Figure 12.) Now, when you create new databases, SQL Server 2005 will use your optimized drive configuration.



Figure 12. Database default locations.

6. To change the tempdb database location to reflect your configuration, open SQL Server Management Studio, connect to your new server, right-click the server, and select New Query. Then, in the query pane, run the commands below. This command will cause SQL Server 2005 to recreate the tempdb data and log files in the new location once you restart of the SQL Server service.



NOTE: The following Transact SQL commands assumes that you want to put your tempdb files in the I:\Tempdb path.

```
USE master  
GO
```

```
ALTER DATABASE tempdb MODIFY FILE  
(  
  NAME = tempdev,  
  FILENAME = 'I:\Tempdb\tempdb.mdf'  
)  
GO
```

```
ALTER DATABASE tempdb MODIFY FILE  
(  
  NAME = templog,  
  FILENAME = 'I:\Tempdb\tempdb.ldf'  
)  
GO
```

7. Restart the SQL Server service using SQL Server Configuration Manager. Click Start | All Programs | Microsoft SQL Server | Configuration Tools | SQL Server Configuration Manager. Then right-click the SQL Server Service, and select Restart.

Configuring SQL Server 2005 via the Surface Area Configuration Tool

The Surface Area Configuration tool provides database administrators and systems administrators a graphical interface for controlling services and features of SQL Server 2005. From the Surface Area Configuration tool, you can start and stop SQL Server-related services. You can also enable or disable many features of SQL Server 2005, including Common Language Runtime (CLR) Integration, Database Mail, xp_cmdshell, or the dedicated administrator console login. The Surface Area Configuration utility also provides a link to the User Provisioning Tool, because users of the Windows 2008 local administrators group do not automatically have the SQL Server sysadmin fixed server role after a default install of SQL Server 2005. To add local Windows administrators to the SQL Server sysadmin role, click Add New Administrator.

Next steps: Best practices for administering and monitoring SQL Server 2005

The optimal performance of your database server depends on more than a successful installation. The run parameters for SQL Server 2005 and the database layout require much careful thought. In this section, we present some best practices that can help improve performance and reliability.

Administering SQL Server 2005 in this environment

In this section, we discuss the following tips in greater detail:

- Select your AUTOGROWTH sizes appropriately: small enough to minimize the effect on performance, but large enough to minimize fragmentation.
- Do not use AUTOGROWTH as a substitute for sizing your data files appropriately.
- Leave AUTO CLOSE at its default value of False.
- Create one tempdb data file per core.
- Leave processor and memory parameters at their defaults unless you have a specific reason for changing them.

AUTOGROWTH

To the greatest degree reasonably possible, size all data, tempdb, and log files appropriately. Ideally, plan file growth and manually expand files during periods when the server is relatively idle. Do not depend on AUTOGROWTH to size files for you. Using AUTOGROWTH can create file fragmentation and negatively affect system performance. While instant file initialization can help reduce the negative performance effect of the AUTOGROWTH feature, it does not eliminate the problem. In addition, instant file initialization does not help reduce file fragmentation at all. Use AUTOGROWTH only as a safety net to prevent data files from filling up and forcing the database to a read-only state.

The AUTOGROWTH increment should be both small enough to limit the performance effect of unplanned file growth and large enough to prevent excessive file fragmentation. While it is impossible to recommend a specific size that will work with all applications, we can suggest that you set the increment using fixed sizes, either in megabytes or gigabytes, rather than percentages. As a file grows, a given percentage of its size also grows in absolute terms, so using percentage increments could lead to uncontrolled file growth and an unacceptable impact on performance.

Even when you have set the AUTOGROWTH increment appropriately, manually growing the files during periods of low activity is still preferable.



NOTE: You can modify AUTOGROWTH settings in the SQL Server Management Studio by right-clicking on the database in question and selecting Properties. Click Files, and scroll to the AUTOGROWTH column.

AUTO CLOSE

Leave the AUTO CLOSE option at its default value of False. For a database in frequent use, AUTO CLOSE can cause many unnecessary opens and closes, which can degrade performance.



NOTE: You can modify AUTO CLOSE settings in the SQL Server Management Studio by right-clicking on the database in question and selecting Properties. Click Options, and scroll to the AUTO CLOSE row.

One tempdb data file per core

When you install SQL Server 2005, the installation process creates a single tempdb primary data file. Running with a single tempdb data file, however, can create unacceptable latch contention and I/O performance bottlenecks. To mitigate these problems, allocate one tempdb data file per processor core.

To check the level of contention you are experiencing in this area, follow these steps:

1. Start the Reliability and Performance Monitor.
2. Click Performance Monitor, right-click the right graph pane, and select Add counters.
3. Scroll down until you find SQLServer:Latches. There are a number of latch-related counters from which to choose.

By default, AUTOGROWTH is set to TRUE for tempdb files. As with data files, however, expanding tempdb too frequently can cause performance to degrade. Therefore, we again recommend allocating the tempdb files enough initial space to accommodate the workload you expect. As a safety net, set the file growth increment large enough to minimize tempdb expansions. By default, the FILEGROWTH increment is 10 percent for the tempdb file, with an unrestricted growth increment of 10 percent. Although we recommend a fixed increment for data files, Microsoft recommends the 10 percent increment for tempdb.

To better utilize the allocation mechanism, make all tempdb data files equal in size. Because SQL Server uses a proportional fill algorithm, different sized files do not distribute the I/O load evenly.



NOTE: You can add or modify tempdb data files and AUTOGROWTH options in the SQL Server Management Studio by right-clicking the tempdb database, selecting Properties, and then clicking Files.

Processor and memory parameters

Leave the processor and memory parameters for SQL Server at their default settings. The default settings run SQL Server at a standard priority, which lets it use all the processors in the server and also makes available as much RAM as SQL Server needs. If

you have special needs, you have the option of changing these settings. The defaults, however, work best for most installations.

Similarly, leave the network packet size at the default value of 4096 bytes. In some cases, such as bulk copy operations or large image files, increasing the packet size can improve efficiency. You should not, however, change the packet size unless you are sure that doing so will improve efficiency.

Finally, whenever possible, run SQL Server 2005 on dedicated servers. With the exception of necessary utilities, run no applications on your server other than SQL Server. Other applications will hurt performance by competing with SQL Server for resources. Also, each additional application makes it harder to tune the server for optimal database performance and to troubleshoot problems.

Monitoring SQL Server 2005 in this environment

Once the server is up and running, you'll want to monitor its performance so you can better tune it for your specific workload. In this section, we discuss some traditional performance monitoring options and SQL Server 2005's Dynamic Management Views and Functions.

Many statistics are available from both Windows and SQL Server. We restrict our discussion to a few that we have found particularly useful. We begin with counters from the OS.

Counters available from the Reliability and Performance Monitor

To see the performance statistics available at the operating system level in Windows Server 2008, follow these steps:

- 1.** Launch the Reliability and Performance monitor, and click Performance Monitor.
- 2.** Right-click the right pane of Performance monitor, and select Add counters.
- 3.** Scroll down to the applicable counter, and expand it.
- 4.** Click the relevant statistic.
- 5.** Select from the list, click Add>>, then click OK.

Figure 13 presents the data ranges we recommend for the following Performance Monitor counters: PhysicalDisk, Processor, and SQLServer: BufferManager.

Counter	Statistic	Explanation
PhysicalDisk	% idle time	Indicator of I/O performance; ideally > 70%
Processor	% processor time	Ideally < 70%
SQLServer: BufferManager	Buffer cache hit ratio	Indicator of how much SQL Server is utilizing physical memory to retrieve data; ideally > 95%

Figure 13. The ranges we recommend for select Performance Monitor counters.

Dynamic management views and functions

Dynamic management views and functions (DMV/DMFs) are powerful tools in SQL Server 2005 that expose the current state of the SQL Server system. You can use the statistics you collect from a DMV/DMF query to analyze server health and diagnose potential problems. Some of these statistics apply to the entire server, while others apply to specific databases. SQL Server 2005 offers over 70 DMV/DMFs.

Some of the more interesting DMV/DMFs include the following:

- **sys.dm_db_index_physical_stats.** This function returns physical characteristics of indexes, e.g., fragmentation percentage, page counts, and row counts.
- **sys.dm_db_index_usage_stats.** This view returns information on different types of index operations (scan, seek, etc.), such as when the system last performed them and how many times it did so.
- **sys.dm_os_performance_counters.** This view returns the current value of some SQL Server performance counters the server maintains. (See Figure 14.)
- **sys.dm_io_virtual_file_stats.** This function returns the I/O statistics for data and log files.

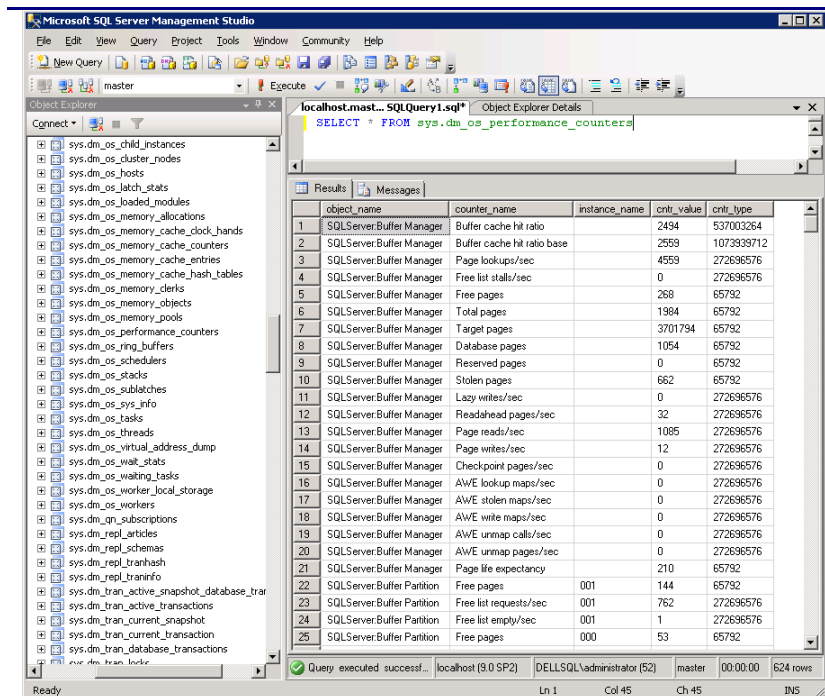


Figure 14. Sample results from the following query: `select * from sys.dm_os_performance_counters`

To query a DMV, use the following steps:

1. Open SQL Server Management Studio.
2. Enter the appropriate server name, enter credentials as necessary, and click Connect.
3. Expand the Databases folder in the Object Explorer.
4. Right-click a database name, and select New Query.
5. Query the desired Dynamic Management View by running the following statements:

```
SELECT * FROM <Dynamic_Management_View_name>
```

For a DMV example:

```
SELECT * FROM sys.dm_os_performance_counters
```

For a DMF example:

```
SELECT * FROM sys.dm_io_virtual_file_stats (NULL, NULL)
```

For more information on Dynamic Management Views, including query options and required parameters, see Microsoft SQL Server

Books Online at <http://msdn2.microsoft.com/en-us/library/ms188754.aspx>



BEST PRACTICE: In all performance monitoring, capture baseline data, set up a regular monitoring plan, and track performance against your goals.

Summing up

Windows Server 2008 Enterprise x64 and SQL Server 2005 provide many powerful features and enhancements. As this Guide documents, the process of deploying these products on a Dell PowerEdge Server is relatively straightforward; you can perform a basic installation in a couple of hours. Spending some up-front time planning can help you avoid potential problems down the road. For example, thinking through the way you will lay out data files can save significant effort later.

To keep your database server running optimally, you must monitor and respond to the changing demands of the workload. Windows Server 2008 and SQL Server 2005 provide a rich set of tools to help you do so. While a thorough treatment of monitoring a SQL Server 2005 database is beyond the scope of this Guide, we have introduced two tools you will find particularly useful for that purpose: the Windows Server 2008 Reliability and Performance Monitor, and the SQL Server 2005 Dynamic Management Views and Functions. We have also given guidelines for some of the counters that can help you locate performance bottlenecks.

The best practices we describe here should help you get the best performance and reliability from your database server.

To learn more, please explore the following resources:
www.dell.com/sql
www.microsoft.com/windowsserver2008/en/us/default.aspx
www.microsoft.com/sql/default.msp

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, Visual Studio and Windows are registered trademarks and Bitlocker, Hyper-V and Windows PowerShell are trademarks of Microsoft Corporation. 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:

- operating system
- tempdb
- logs
- user data

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.

Note: In the instructions below, we use the names and numbers of the controllers as our test setup displayed them. Thus the PERC5/i showed as Controller 0 and the PERC 5/E showed as Controller 1. The numbering on your system might differ.

- 1.** During the boot sequence, you will see the message "Press <Ctrl><R> to Run Configuration Utility." Press <Ctrl><R>. Note: You only have 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 5/i). Press return.
- 3.** While Controller 0 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 return.
- 6.** A message box appears telling you that initializing the RAID is highly recommended but destructive. Click OK.
- 7.** If you have an available disk to use as a hot spare, follow these steps:
 - a. Highlight Disk Group 0, press F2, and select Manage Ded. HS.
 - b. Press the space bar to select your hot spare disk drive, which the system will dedicate to this Disk Group. Tab to OK, and press Enter.
- 8.** Highlight Virtual Disk 0, and press F2.

- 9.** Select Initialization, then select Fast Init. Click OK when the system prompts you with any warnings.
- 10.** Press F12 to return to the main level of the configuration utility.
- 11.** Select the controller for the drive array (in our case, the PERC 5/E). Press Return.
- 12.** Highlight controller 1, press F2, and select Create New VD.
- 13.** Leave the RAID level at 0, and tab to the Physical Disks section. Select the first two drives (Drives 0 and 1) using the space bar. Tab to highlight OK, and press Return.
- 14.** A message box appears telling you that initializing the RAID is highly recommended but destructive. Make sure OK is selected, and press Enter.
- 15.** Highlight Controller 1 once again, and press F2. Select Create New VD, and press Enter.
- 16.** With the RAID level field highlighted, press Enter to open the drop-down list, and select RAID 1. Tab to the Physical Disks section, and select the first two drives (Drives 2 and 3) 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.
- 17.** A message box appears telling you that initializing the RAID is highly recommended but destructive. Click OK.
- 18.** Once again highlight Controller 1, press F2, highlight Create New VD, and press Enter.
- 19.** Press Enter, and then arrow down to select RAID level 10, and press enter to select RAID 10. Tab to the Physical Disks section, and select all drives (Drives 4 through 13) except the last (Drive 14). Note that the drives you allocated earlier no longer show up in the available list. Tab to highlight OK, and press return.
- 20.** A message box appears telling you that initializing the RAID is highly recommended but destructive. Click OK.
- 21.** To use the remaining disk as a hot spare on the PowerVault MD1000:
 - 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. This disk is now a global hot spare for all three Disk Groups on the external storage array.
- 22.** Press <Ctrl><P> to navigate back to the Virtual Disk Management screen. Highlight each Virtual Disk, and press F2. Select Initialization, then select Fast Init. Accept OK when the system prompts you with a warning. After you select Fast Init, a progress bar appears briefly, then disappears.
- 23.** 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.

- 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. (4 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, if necessary, and click Next
- 5.** At the Select the edition of Windows that you purchased screen, click Windows Server 2008 Enterprise (Full Installation), check the box next to I have selected the edition of Windows that I purchased, 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). Note: The upgrade option should be disabled.
- 8.** In the Where do you want to install Windows? screen, click Drive options (Advanced).
- 9.** Highlight any existing partitions, and click Delete.
- 10.** At the warning screen, click OK.
- 11.** When there are no more partitions, highlight the unallocated space, and click New.
- 12.** Click Apply.
- 13.** Highlight the new partition for the disk on which you plan to install the operating system, and click Format.

- 14.** Click OK.
- 15.** Highlight the partition and click Next to start the installation. (20 minutes, during which Windows restarts twice)
- 16.** When the system indicates that you must change the user's password before you log on the first time, click OK to change the password.
- 17.** Enter a strong password in both the New password and Confirm password fields, and click the arrow.
- 18.** At the Your password has been changed prompt, click OK. Windows then prepares the desktop. Install Windows updates. (2 minutes for us; download times vary)
- 19.** 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. Then click OK.
- 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 Close to close the System Properties window.
- 19.** Click Restart Now to reboot the server.

Configuring data drives

Configure each of the three data (non-operating system) RAID drives using the following process. Note that Windows may not recognize the drives until the hardware RAID utility completes its background check.

- 1.** Click Start.
- 2.** Expand Administrative Tools, and click Computer management.
- 3.** In the left panel, expand Storage, and select Disk Management. If a dialog asking to initialize a disk appears, click OK. You cannot proceed if the disks are not initialized.
- 4.** Right-click each unallocated volume, and select New Simple Volume.
- 5.** At the New Simple Volume Wizard window, click Next.
- 6.** Leave the volume size at the default, which is to use all available space on the volume, and click Next.
- 7.** Assign a drive letter if you want something other than the default (we left the drive letter at its default), and then click Next.
- 8.** Leave the File System as NTFS.
- 9.** Set the Allocation Unit size to 64 K. Windows Server 2008 will gray out the option to compress the drive. You may optionally select the quick format. Click Next.
- 10.** Review your selected settings, and click Finish.

Appendix C. Methodology: Installing SQL Server 2005



NOTE: Because this scenario did not include upgrading from a previous version of SQL Server, we did not run the Upgrade Advisor.

If you plan on running SQL Server 2005 within the context of a domain user account, which we recommend, then have your Active Directory administrator create this account and assign the password. You will use this authentication information during the installation.

Install Microsoft SQL Server 2005 Enterprise by following these steps. We provide approximate times for each group of steps in the first step of that group.

Plan on at least 20 minutes to complete the installation.

1. Insert the SQL Server 2005 DVD into the DVD drive.
2. If the gray splash screen appears, click Server components, tools, Books Online, and samples. If you are installing from an MSDN CD, you may need to navigate to the SQL Server 2005 splash screen.
3. On the End User License Agreement screen, check the I accept the licensing terms and conditions checkbox, and click Next. (7 seconds)
4. On the Installing Prerequisites screen, click Install to begin installing the necessary prerequisites for SQL Server 2005. (less than 1 minute)
5. After the prerequisites installation finishes, click Next.
6. On the Welcome to the Microsoft SQL Server Installation Wizard screen, click Next.
7. The System Configuration Screen appears and checks that the server meets all hardware and software requirements. Note any warnings or errors.



NOTE: SQL Server 2005 Reporting Services has a dependency on Internet Information Services (IIS). Unless you have configured IIS on your server, warning messages regarding the IIS Feature Requirement and the ASP.Net version registration appear. If you are not installing SQL Server Reporting Services, you can ignore these warnings.

- 8.** Enter the appropriate licensing information on the Registration Information screen.
- 9.** On the Feature Selection screen, select only those components you need for this particular installation of SQL Server 2005. This practice reduces overhead on your Dell PowerEdge 2970. If you plan on running only the database engine, then select only Database Engine Services and Workstation Components. The Workstation components are helpful for administering this server locally.
- 10.** Click Next to continue.
- 11.** 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. You may have multiple named instances per server, although the resources SQL Server requires will increase.
- 12.** Click Next to continue.
- 13.** On the Service accounts screen, enter the account name of the account that will run SQL Server 2005 and the password you want to use; ideally, this account would be a domain service account. Next, enter the password, and finally enter the domain. If you want to customize each service to use separate accounts, check the Customize for each service account checkbox. Click Next.
- 14.** On the Authentication Mode screen, select which authentication mode you want to run. We recommend Windows Authentication Mode. If you select Mixed Mode, you must enter a password for the sa account. Click Next.
- 15.** On the Collation screen, accept the defaults unless you have a specific requirement for other collations, and click Next to continue.
- 16.** On the Error and Usage Report Settings, leave the checkboxes unchecked, unless you need to send usage and error reports to Microsoft or another reporting server.
- 17.** The Ready to install screen offers one last chance to review your choices. After you do so, click Install. (8 minutes)
- 18.** If you receive a warning regarding compatibility issues between Visual Studio 2005 and this version of Windows, click Run Program to continue the installation process.
- 19.** After all components are complete, click Next.

- 20.** At the Completing Microsoft SQL Server 2005 Setup screen, a hyperlink to the Summary log file appears for setup troubleshooting (the default location is C:\Program Files\Microsoft SQL Server\90\Setup Bootstrap\LOG). Click Finish to complete the installation.
- 21.** After the SQL Server 2005 installation process completes, check the Microsoft Downloads Center (<http://www.microsoft.com/downloads>) for the latest SQL Server updates, patches, and service packs. We used SQL Server 2005 SP2, the latest update available from the Microsoft downloads site when we wrote this manual. Note that the supported version of SQL Server 2005 on Windows Server 2008 is Service Pack 2. Install the latest Service Pack after downloading. (download time is about 35 minutes; installation time is 4 minutes)

 - a. If you receive a message that some processes are locked, you can either stop the process using SQL Server Configuration Manager to avoid a restart of your system, or simply continue with the SP installation if a restart is acceptable after the SP installation completes. If you receive a prompt to reboot the server, click OK.
 - b. If you receive a warning regarding compatibility issues between Visual Studio 2005 and this version of Windows, click Run Program to continue the installation process.
- 22.** At the conclusion of the installation of SP2, the installation software allows you to launch the User Provisioning Tool. This program lets you provide local Windows administrator users full access to SQL Server 2005. Because the SP2 installation process might stop certain SQL related services, we found it advisable to reboot your server before you run the User Provisioning Tool. To do so, uncheck the option to "Launch the User Provisioning Tool for Windows Vista after SP2 installation completes," and then reboot the server. If you want to run the provisioning tool manually after rebooting, you will find a link to the user provisioning tool within the SQL Server Surface Area Configuration Tool.
- 23.** If you need to create a User database, follow the steps below. Note: Many applications that use SQL Server as a backend database automatically create their own user databases as part of the installation program.
- 24.** Open SQL Server Management Studio.

- 25.** Enter the appropriate server name, enter credentials as necessary, and click Connect.
- 26.** Right-click Databases in Object Explorer, and select New Database.
- 27.** Enter the database name.
- 28.** Leave the owner as <default>.
- 29.** If needed, scroll to the right of the Database Files window, and enter the appropriate paths for the database file and log file. If you did not modify the default locations on the Database Settings screen of the Server Properties after installation, the default locations for these paths are with the SQL installation directories, typically the C: drive. If you modified the default locations, then these paths should reflect your preferences.
- 30.** Click OK to create the database.

Using the ALTER DATABASE command to modify the location of user database or tempdb files and logs after an installation

In this example, we called the user database Userdb. We named the data file Userdb.mdf and the log file Userdb_log.ldf. Be sure you have exclusive access to any database you plan to move. Also, be sure you have given full control of any folders you use to the SQL Server service account, as we explained in the section Changing tempdb and default user database file locations above.

- 1.** Open SQL Server Management Studio.
- 2.** Enter the appropriate server name, enter credentials as necessary, and click Connect.
- 3.** Expand the Databases view in the Object Explorer. Right-click the database name whose file location you wish to change, and select Properties. Note: System databases such as tempdb do not appear in the Userdb view. You can find them in the System Databases view under the Databases view. Under Select a page, select Files, and take note of the logical name, path, and file name of the items you plan to move.
- 4.** Click OK to close the Properties dialog. Right-click the database name whose file location you wish to change, and

select Tasks | Take Offline. Note: This step is unnecessary when you're moving tempdb files or tempdb logs.

- 5.** When the Take Offline command completes, click Close to close the dialog box. Note: This step is unnecessary when you're moving tempdb files or tempdb logs.
- 6.** Use Windows Explorer to move the file or files to the new location. Note: This step is unnecessary when you're moving tempdb files or tempdb logs. SQL Server automatically recreates the tempdb files and tempdb logs in the new location after the service restarts, so there is no need to physically move tempdb files.
- 7.** Under Databases | System databases, right-click Master, and select New Query.
- 8.** For each user database data file (.MDF, .NDF), tempdb data file (.MDF, .NDF), or log file (.LDF) that you want to relocate, enter the following statements in the query window:

```
USE master
GO
```

```
ALTER DATABASE database_name MODIFY FILE
(
    NAME = logical_name,
    FILENAME = 'new_path\os_file_name'
)
GO
```

For example:

```
USE master
GO

ALTER DATABASE Userdb MODIFY FILE
(
    NAME = Userdb,
    FILENAME = 'K:\SQLData\ Userdb.mdf'
)
GO

ALTER DATABASE Userdb MODIFY FILE
(
    NAME = Userdb _log,
    FILENAME = 'J:\SQLLogs\ Userdb.ldf'
)
GO
```

9. After entering the SQL commands in a query window, click the Execute button to run them.
10. If you are changing a user database, right-click the database name whose file location you changed, and select Tasks | Bring Online. Note: If you're moving tempdb files or logs, you must stop and restart SQL Server at this point.
11. When the Bring Online command completes, click Close to close the dialog box. Note: This step is unnecessary when you're moving tempdb data files or tempdb log files.
12. You can verify the file change by typing the following statements in a query window:

```
Use database_name
exec sp_helpfile
```

For example:

```
USE Userdb
exec sp_helpfile
```



BEST PRACTICE: After you verify that SQL Server has successfully recreated the tempdb data files and tempdb log files, delete the old files. Doing so will free disk space and avoid potential later confusion.

About Principled Technologies

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 research into new technologies, to the development of new methodologies, to testing with existing and new tools.



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

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.