DEPLOYMENT GUIDE

SQL SERVER™ 2008 x64 ON WINDOWS SERVER™ 2003 R2 ENTERPRISE x64 ON DELL POWEREDGE™ 2950

Table of contents

Introduction

Dell PowerEdge 2950

The Dell™ PowerEdge™ 2950 is an excellent server for deploying SQL Server 2008 x64, the latest release of Microsoft's highly available and high-performing data management platform. This Deployment Guide reviews the approach that Principled Technologies (PT) and Dell recommend for implementing a tested and validated solution for SQL Server 2008 x64 on the Dell PowerEdge 2950 9th generation server and Microsoft Windows Server 2003 Enterprise R2 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 new features in SQL Server 2008 x64.

New features of SQL Server 2008 x64

SQL Server 2008 x64 introduces a wide variety of new features. While there are far too many to discuss in this Guide, the following list mentions some of the more significant ones:

- **Policy-based management.** SQL Server 2008 x64 lets you automatically enforce and monitor policies for database operations. You can also push these policies out to your servers.
- **Resource Monitor.** The Resource Monitor allows you to monitor which users are running which queries and how long those queries run. This ability helps you to better tune your server for optimal performance.
- **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 x64 can automatically compress database backups, a feature previously only available via the purchase of third-party software products.
- **Performance data collection.** SQL Server 2008 x64 lets you store performance data in a centralized database. It also provides enhanced tools for analyzing and reporting on such performance data.

- **FILESTREAM data.** The new SQL Server 2008 x64 FILESTREAM data type allows you to store large binary data, such as documents or images, directly in an NTFS file system. The data object remains part of the database and maintains transactional consistency. Applications can access the data via an NTFS streaming API.
- **"Hot add" CPUs.** On supported hardware platforms you can now add one or more CPUs while SQL Server 2008 x64 is running. As with SQL Server 2005, you can also hot add memory.

Installing and setting up Windows Server 2003 R2 Enterprise x64 and SQL Server 2008 x64 in this environment

In this section, we focus on installing and setting up Windows Server 2003 R2 Enterprise x64 and SQL Server 2008 x64 on the Dell PowerEdge 2950 server. We begin by defining the environment we used for our sample setup and then address configuring the server's drives.

Defining our environment

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 in the Solutions Deliverable List (SDL) at www.dell.com/sql.

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 2950 with the following components:

- two Intel Xeon Processor E5335 (Clovertown) processors, each running at 2.0 GHz
- 16 GB of PC2-5300F RAM
- three 73.0 GB, 15,000 rpm, Seagate Cheetah ST373455SS SAS hard disks
- Dell PERC 5/i integrated RAID controller
- two Broadcom BCM5708C NetXtreme II GigE integrated NICs

We connected the server to an external Active Directory domain controller via a gigabit switch.

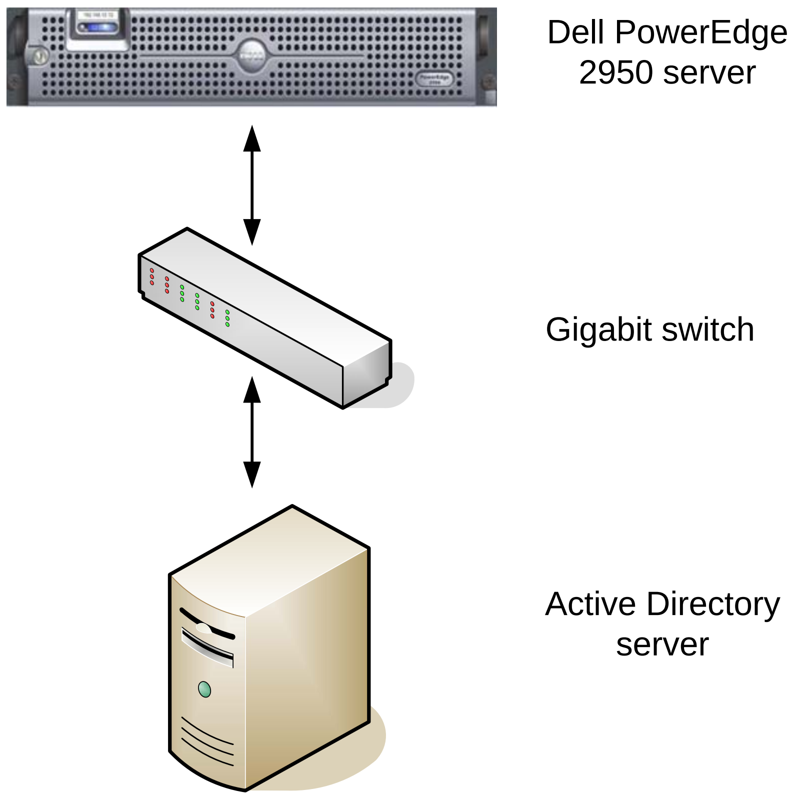


Figure 1. The setup we used in our hands-on testing and research for this Deployment Guide.

BEST PRACTICE: Deploy SQL Server 2008 x64 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.

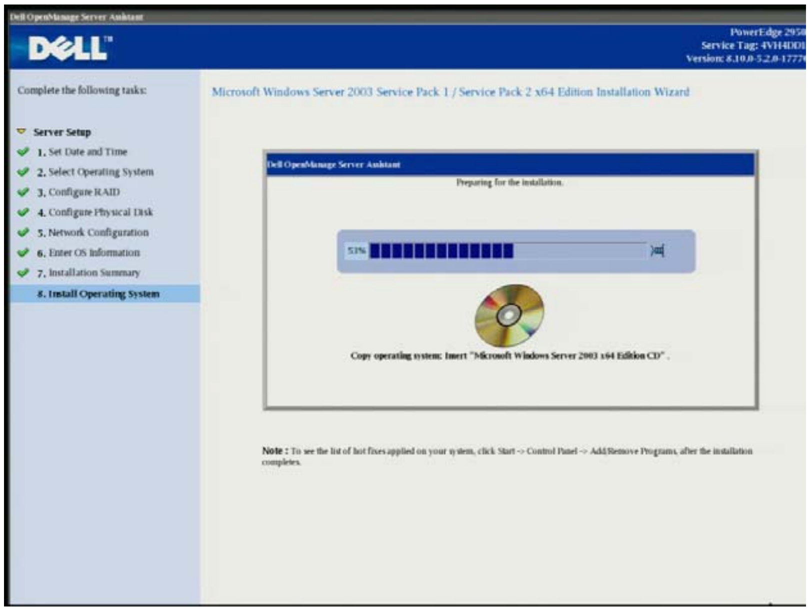
We installed Windows Server 2003 R2 Enterprise x64 using the Dell OpenManage Server Assistant 8.10.0 from the Dell PowerEdge Installation and Server Management Bootable CD version 5.2. Then we applied Service Pack 2 before installing SQL Server 2008 x64. The version of SQL Server we used was the November Community Technology Preview (CTP), which identified itself as: Microsoft SQL Server code name “Katmai” (CTP) - 10.0.1075.23 (X64) Nov 8 2007 14:13:37. This was the latest version of SQL Server 2008 x64 available as of the start of our work on this Guide.

Installing Windows Server 2003 R2 Enterprise x64

This section provides an overview of the operating system installation process. We include approximate wait times for each step.

NOTE: Plan on at least 90 minutes for installing Windows Server 2003 R2 Enterprise x64 on the Dell PowerEdge 2950 server. Each step below takes at least a minute. We put the approximate amount of time each step takes in parentheses at the end of the description of that step. Those times below exclude data entry time. The time to install updates, which was 10 minutes in our setup, will increase over time as Microsoft releases additional OS updates.

1. Insert the Dell PowerEdge Installation and Server Management bootable CD version 5.2, and reboot the system. Note: During this process, the software will ask if you want to insert a replication floppy; we did not. (5 minutes, during which the system boots twice.)
2. On the Welcome to the Dell OpenManage Server Assistant screen, click Cick Here for Server Setup. Set the date and time. Select Microsoft Windows Server 2003 Service Pack 1/Service Pack 2 x64 Edition. Skip RAID configuration. Set the boot partition size to the maximum available – 69335 MB and leave the File system at NTFS. Enter the IP addresses and select the correct subnet type. Enter your configuration information. (6 minutes)


Dell OpenManage Server Assistant
DELL
PowerEdge 2950
Service Tag: 4VH4DD1
Version: 8.10.0-5.2.0-1777
Complete the following tasks:
Server Setup
1. Set Date and Time
2. Select Operating System
3. Configure RAID
4. Configure Physical Disk
5. Network Configuration
6. Enter OS Information
7. Installation Summary
8. Install Operating System
Dell OpenManage Server Assistant
53%

Microsoft Windows Server 2003 Service Pack 1 / Service Pack 2 x64 Edition Installation Wizard

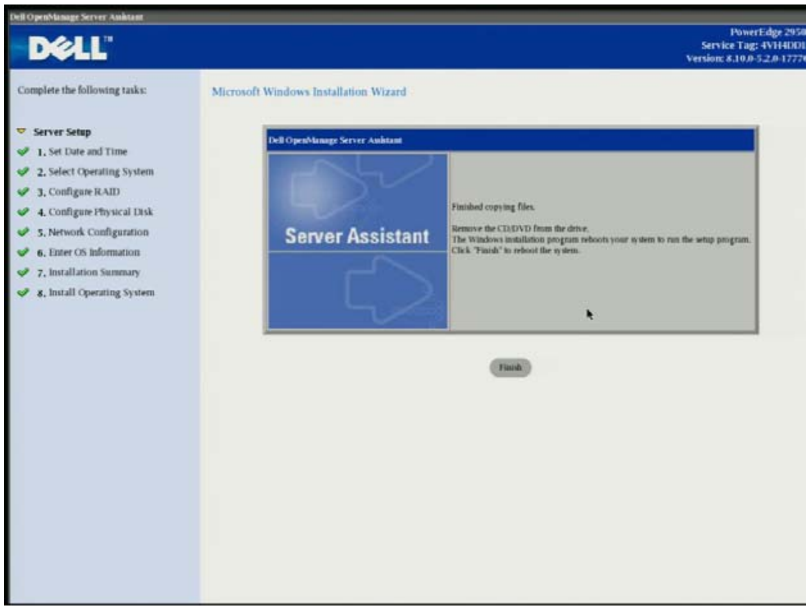
Preparing for the installation.

Copy operating system: Insert "Microsoft Windows Server 2003 x64 Edition CD" .

Note : To see the list of hot fixes applied on your system, click Start -> Control Panel -> Add/Remove Programs, after the installation completes.

Figure 2. Dell OpenManage Server asking for the Windows installation CD.

3. When prompted, insert the Microsoft Windows Server 2003 CD. (5 minutes)


Dell OpenManage Server Assistant

PowerEdge 295
Service Tag: 4VH4DD
Version: 8.10.0-5.2.0-1777

Complete the following tasks:

- Server Setup
- 1. Set Date and Time
- 2. Select Operating System
- 3. Configure RAID
- 4. Configure Physical Disk
- 5. Network Configuration
- 6. Enter OS Information
- 7. Installation Summary
- 8. Install Operating System

Microsoft Windows Installation Wizard

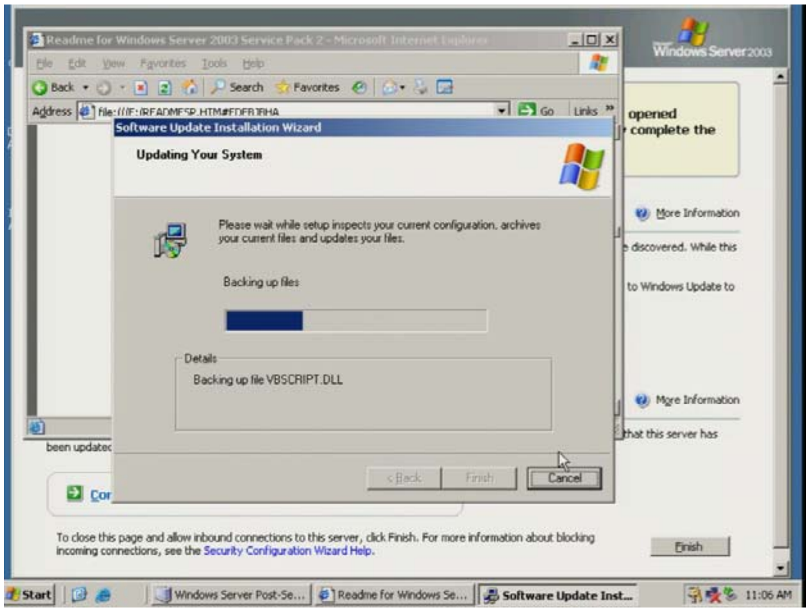
Dell OpenManage Server Assistant

Finished copying files.

Remove the CD/DVD from the drive.
The Windows installation program reboots your system to run the setup program.
Click "Finish" to reboot the system.

Figure 3. Operating System Files copied, ready to start the install.

4. When prompted, remove the CD from the system. Click Finish. The installation begins. (27 minutes, during which the system reboots 4 times.)
5. When the system comes available, log in as Administrator, and insert the second Windows server 2003 CD. Insert it and click OK. (2 minutes)
6. Install SP2. We did this from a CD, accepting the license agreement and the default options. (3 minutes)



Please wait while setup inspects your current configuration, archives your current files and updates your files.

Backing up files

Details

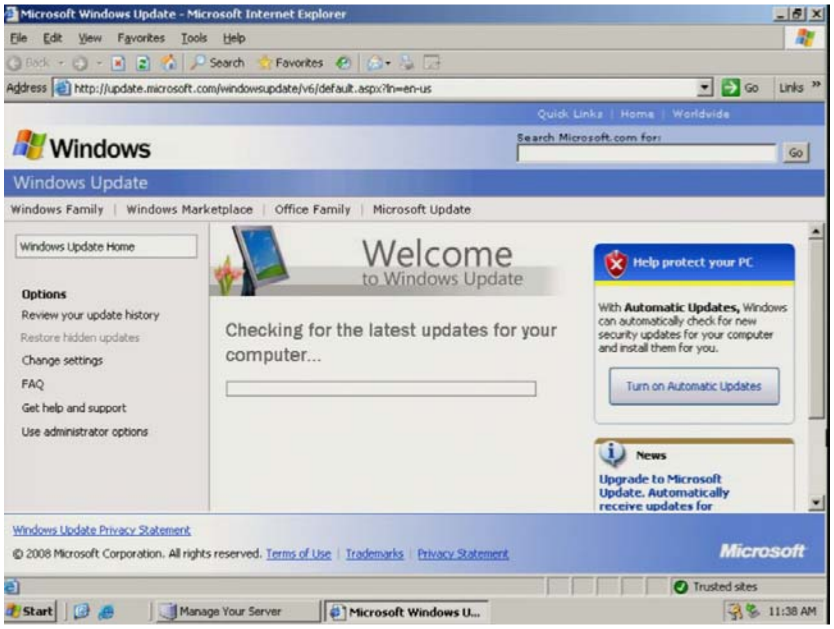
Backing up file VBSCRIPT.DLL

To close this page and allow inbound connections to this server, click Finish. For more information about blocking incoming connections, see the Security Configuration Wizard Help.

Figure 4. Installing Service Pack 2.

7. Reboot system (5 minutes)

8. Run Windows update. We installed all essential updates. When the installation software prompted us to install Internet Explorer 7, we declined. (Note: The time will vary; in our case, it was 10 minutes.)



Address http://update.microsoft.com/windowsupdate/v6/default.aspx?ln=en-us

Checking for the latest updates for your computer...

With Automatic Updates, Windows can automatically check for new security updates for your computer and install them for you.

Upgrade to Microsoft Update. Automatically receive updates for

© 2008 Microsoft Corporation. All rights reserved. Terms of Use | Trademarks | Privacy Statement

Figure 5. Windows updates examining the server.

BEST PRACTICE: Configure all database servers with static IP addresses. This practice assures that SQL Server 2008 x64 resources will remain available even in the event of a DHCP server failure. It also increases the stability of your networking and DNS environments.

BEST PRACTICE: Use CNAME records to assign aliases to your database servers. This practice 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 drives

The next step is to configure the remaining drives in your server. Format all SQL Server volumes, including those you use to hold your data, logs, and temporary database (tempdb) as NTFS file system. Microsoft recommends the allocation unit size for these volumes be 64 KB. Avoid values less than 8 KB, as they can increase the risk of torn pages (i.e., pages whose contents are split across disk allocation units). Appendix A presents detailed drive configuration steps.

NOTE: Ideally, the operating system would also be on its own drive. In this three-drive case, we must either have two of the SQL Server components share a drive or put a component on the drive with the operating system. We chose to put the log files on the drive with the operating system. The operating system installation sets the allocation unit size to 4 KB, instead of the 64 KB Microsoft recommends. While it is possible to use the recovery console to format the OS drive to a 64 KB allocation unit size, we do not recommend this practice due to the potential negative effect on the performance of OS functions.

Do not use compressed drives. Windows Server 2003 Enterprise R2 x64 does not offer the option to compress drives when the allocation unit size is 64 KB. Moreover, SQL Server 2008 x64 will create only read-only databases on compressed drives, which makes such drives useful only for historical databases.

Fortunately, SQL Server includes built-in compression, which we discuss below. Use SQL Server's built-in compression when your blend of performance and capacity needs suggest that it makes sense to do so.

For the discussion below, we will view SQL Server 2008 x64 as using three specific storage areas: the temporary database file (tempdb), the log files, and the data files for databases you create. We make this division based on the different characteristics of these types of files. The actual situation is more complicated, of course: the tempdb is one of several system databases, albeit the most volatile one, and there is at least one log file for every database.

One part of the installation process for SQL Server 2008 x64 is creating the tempdb system database, which all databases on the server share. SQL Server uses the storage of the tempdb database for such purposes as sorting, building aggregates for GROUP BY or ORDER BY clauses, composing the temporary tables resulting from JOIN statements, querying using DISTINCT, and storing cursors.

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

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

BEST PRACTICE: Separate the tempdb and transaction log files onto their own disks on separate disk groups when possible. Doing so can help 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 or a server with few disks (as in this Guide), such total isolation is not possible.

Likewise, when possible, create full-text catalogs on their own physical drive(s).

For consolidation, 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.

To achieve optimal performance that scales with heavy workloads, Microsoft recommends that the number of data files for a SQL Server 2008 x64 database range from 0.5 to 1 for each CPU core on the server. You do, however, pay a cost for having multiple files in a filegroup. If these files share spindles, the server can experience contention when multiple database processes attempt to access them simultaneously. Therefore, you must balance the possibility of contention for multiple files on a spindle against the demand for data to satisfy the processor cores. In this case, consider breaking the data files into a number of files equal to half the number of cores. In the case of the tempdb, use one data file per core.

NOTE: You can add or modify data files and filegroups in the SQL Server Management Studio by right-clicking on the database in question, selecting Properties, and then clicking Files or Filegroups.

Installing SQL Server 2008 x64

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.

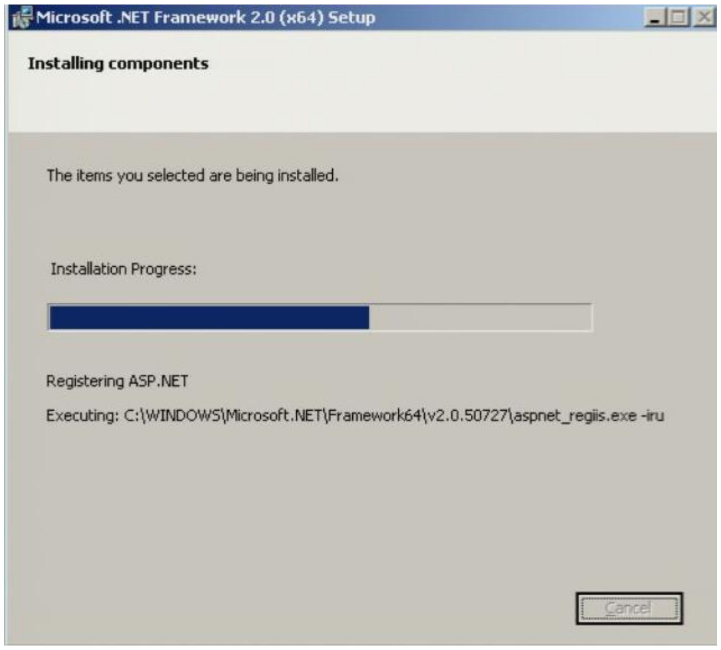
You should, however, deploy SQL Server on a member server in an Active Directory domain. Do not make the SQL Server service domain accounts members of the Domain Administrators group. In fact, grant only the necessary rights on the local server to the SQL Server service account as part of your pre-installation 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 Books Online at msdn2.microsoft.com/en-us/library/ms143504(SQL.100).aspx#Review_NT_rights.

This section provides an overview of the SQL Server 2008 x64 installation process. We include approximate wait times for each step. Appendix B provides full, detailed installation instructions. Appendix B also includes instructions for creating a user database and instructions for using the ALTER DATABASE command to move an existing database.

NOTES: We installed SQL Server 2008 using an ISO image we burned to a DVD. Microsoft's download page for the November 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 x64 on the Dell PowerEdge 2950 server. Each step below takes at least a minute. We put estimated times in parentheses at the end of each step. Those times exclude data entry time.

1. Insert the SQL Server 2008 x64 DVD into the DVD drive. Click Server components, Tools, Books Online, and Samples.
2. Accept the license agreement for .Net 2.0 and install it. (6 minutes)



The items you selected are being installed.

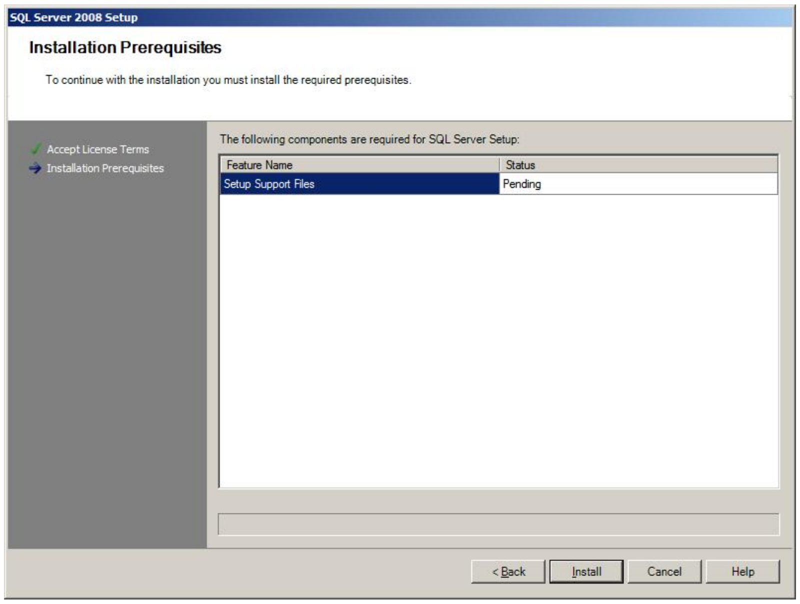
Installation Progress:

Registering ASP.NET

Executing: C:\WINDOWS\Microsoft.NET\Framework64\v2.0.50727\aspnet_regiis.exe -iru

Figure 6. Installing .NET Framework 2.0.

3. Accept the license agreement; and install the prerequisites. (1 minute)



SQL Server 2008 Setup

Installation Prerequisites

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

- ✓ Accept License Terms
- → Installation Prerequisites

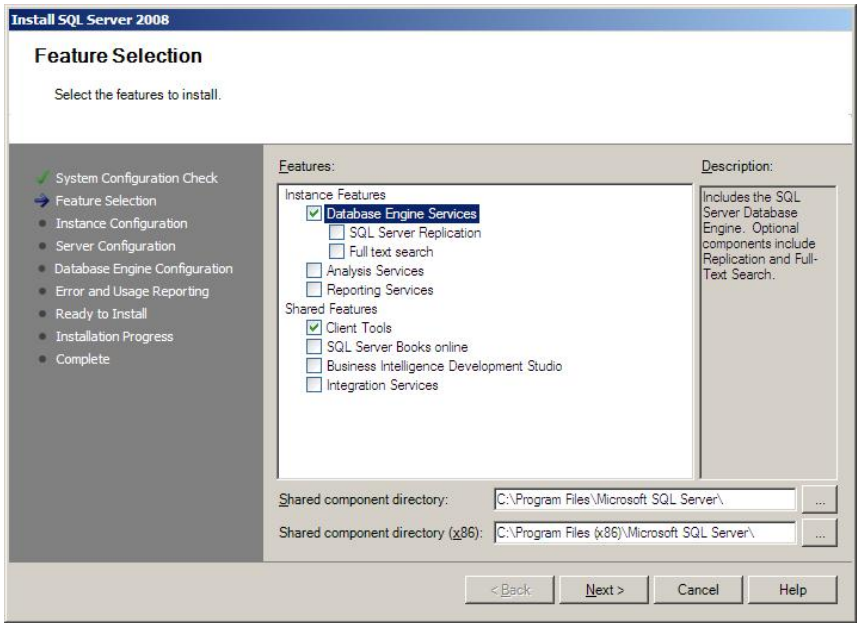
The following components are required for SQL Server Setup:

Feature Name	Status
Setup Support Files	Pending

< Back | Install | Cancel | Help

Figure 7. Installing SQL Server 2008 x64 prerequisites.

4. 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. You can install additional components later as necessary.



Install SQL Server 2008

Feature Selection

Select the features to install.

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

Features:

- Instance Features
  - [✓] Database Engine Services
    - [ ] SQL Server Replication
    - [ ] Full text search
  - [ ] Analysis Services
  - [ ] Reporting Services
- Shared Features
  - [✓] Client Tools
  - [ ] SQL Server Books online
  - [ ] Business Intelligence Development Studio
  - [ ] Integration Services

Description:

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

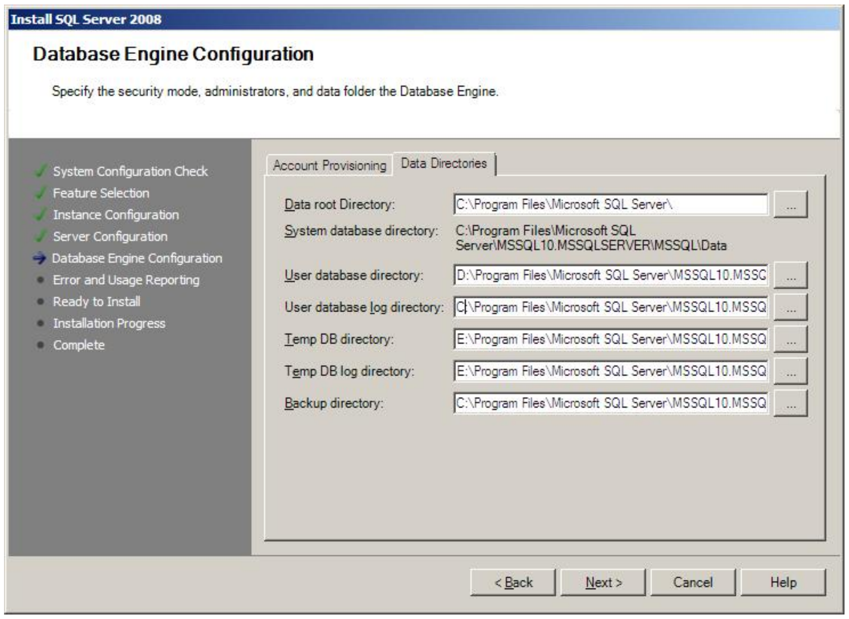
Shared component directory: C:\Program Files\Microsoft SQL Server\

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

< Back | Next > | Cancel | Help

Figure 8. Selecting SQL Server 2008 x64 features.

5. On the following screens, enter the account information and data file locations. Install. At the confirmation screen, click Next. (11 minutes)



Database Engine Configuration

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

Data root Directory:	C:\Program Files\Microsoft SQL Server\
System database directory:	C:\Program Files\Microsoft SQL Server\MSSQL10.MSSQLSERVER\MSSQL\Data
User database directory:	D:\Program Files\Microsoft SQL Server\MSSQL10.MSSC
User database log directory:	C:\Program Files\Microsoft SQL Server\MSSQL10.MSSQ
Temp DB directory:	E:\Program Files\Microsoft SQL Server\MSSQL10.MSSQ
Temp DB log directory:	E:\Program Files\Microsoft SQL Server\MSSQL10.MSSQ
Backup directory:	C:\Program Files\Microsoft SQL Server\MSSQL10.MSSQ

Figure 9. Setting file locations.

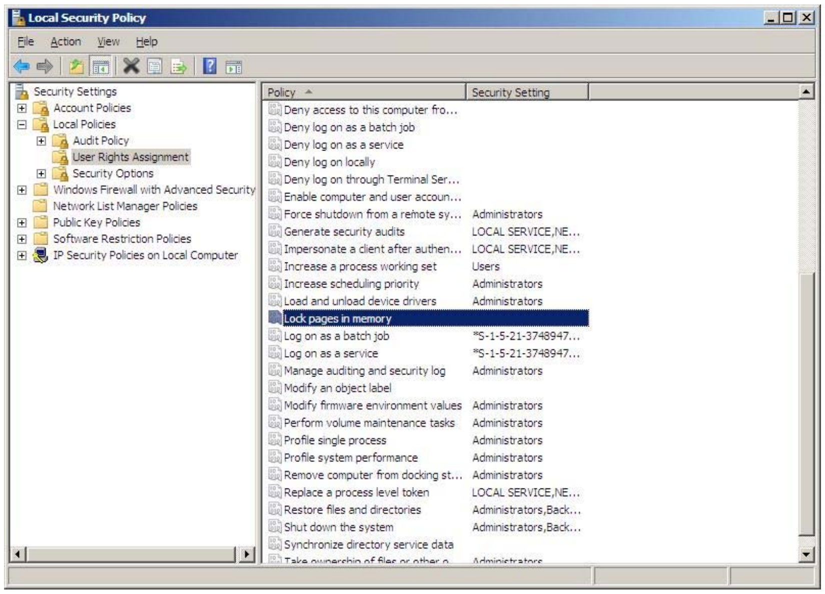
6. At the Complete screen, note the Summary log file location, and click Close to complete the install.

Because we were installing a Community Technology Preview (CTP) release of SQL Server 2008 x64, 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 x64, after the installation process completes you should check the Microsoft Downloads Center (http://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 will dynamically allocate and deallocate 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 2003 Enterprise R2 x64 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 x64 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 Click User Rights Assignment.
4. Scroll down to Lock pages in memory, and double-click it. Click Add User or Group, and enter the SQL Server service account name.



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

Figure 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 2008 x64 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.

BEST PRACTICE: 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 will not initialize storage it allocates. Instead, the storage will remain 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 x64 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 it.
5. Click Add User or Group, and enter the account name.

Next steps: Best practices for administering and monitoring SQL Server 2008 x64

The optimal performance of your database server depends on more than a successful installation. The run parameters for SQL Server 2008 x64 and the database layout require much careful thought. In this section, we present a few best practices that will improve performance and reliability. A detailed discussion of such practices may follow in a later Guide.

Administering SQL Server 2008 x64 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 affect system performance. (File expansion always affects 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 will 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 recommend 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.

Log files raise an additional issue. Within a physical log file, there are multiple virtual log files. A virtual log cannot span file extents. Thus, if you set the AUTOGROWTH increment at 5MB, you end up limiting the maximum size of a virtual log to 5 MB, which can make certain large transactions impossible to complete.

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 2008 x64, 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. (This is different than the practice for user-defined database files, where we recommend 0.5 to 1 data file per 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, then 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 expected workload. 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, differently 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 on 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 2008 x64 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 or to troubleshoot problems.

Notes on the new SQL Server 2008 x64 compression capabilities

As we explained earlier, for SQL Server 2008 x64 installations never use the disk compression feature of Windows Server. In fact, SQL Server 2008 x64 will let you create read-only databases on compressed drives. (That ability can, of course, prove useful for historical databases). Moreover, Windows Server 2003 grays out the option to compress a drive when the allocation unit size is set to 64KB, which is the size we recommend above.

That said, SQL Server includes its own built-in compression options and there are circumstances where using it makes very good sense.

SQL Server 2008 x64 actually provides two compression features: backup compression and data compression. Backup compression uses standard file compression techniques. Early benchmarking has demonstrated compression ratios in the 5:1 range, though, as always, the compression level you achieve will depend on your data set. Use compressed backups if they make sense for your operation.

In SQL Server 2008 x64, the Storage Engine transparently manages compression at the row level and the page level. (Because pages contain rows, page-level compression implies row-level compression.)

You can enable compression on a table or index, or even a part of a table or index. As with any compression, the compression ratio and the CPU overhead will depend on the characteristics of your particular data.

CAUTION: Do not compress log files. Doing so can negatively affect both performance and reliability.

Monitoring SQL Server 2008 x64 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 2008 x64's new Performance Data Collector.

Many statistics are available from both Windows and SQL Server. We restrict our discussion to a few that we have found particularly useful. (More detailed discussions of monitoring may be the topic of a future Guide.) We begin with counters from the OS.

Counters available from Performance Monitor

To see the performance statistics available at the operating system level in Windows Server 2003, follow these steps:

1. Click Start
2. Go to Administrative Tools | Performance.
3. Right-click the right pane of Performance Monitor, and select Add counters.
4. Select the relevant Performance Object from the drop down list.
5. Scroll down to the relevant statistic.
6. Select from the list, click Add, and then click Close.

Counter	Statistic	Explanation
PhysicalDisk	% idle time	Indicator of I/O performance; ideally > 20%
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%

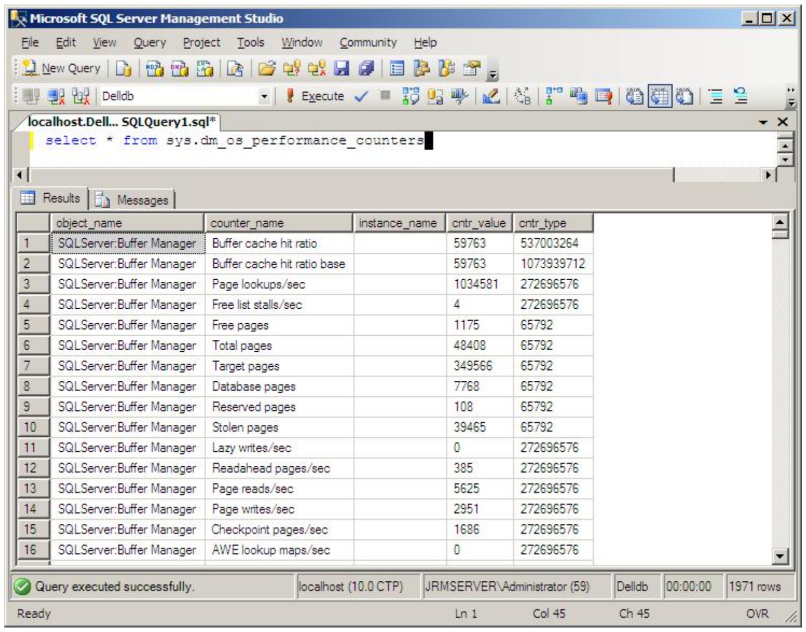
Figure 11. Data ranges we recommend for select Performance Monitor counters.

Dynamic management views and functions

Dynamic management views and functions (DMV/DMFs) 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 2008 x64 offers over 100 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.
- **sys.dm_io_virtual_file_stats.** This function returns the I/O statistics for data and log files.
- **sys.dm_resource_governor_resource_pools.** This new SQL Server 2008 x64 view returns information about the current resource pool state, the configuration of resource pools, and other resource pool statistics.


Microsoft SQL Server Management Studio
File Edit View Query Project Tools Window Community Help
New Query
Delldb
Execute
localhost.Dell... SQLQuery1.sql*
Results
Messages

```
select * from sys.dm_os_performance_counters
```

	object_name	counter_name	instance_name	cntr_value	cntr_type
1	SQLServer:Buffer Manager	Buffer cache hit ratio		59763	537003264
2	SQLServer:Buffer Manager	Buffer cache hit ratio base		59763	1073939712
3	SQLServer:Buffer Manager	Page lookups/sec		1034581	272696576
4	SQLServer:Buffer Manager	Free list stalls/sec		4	272696576
5	SQLServer:Buffer Manager	Free pages		1175	65792
6	SQLServer:Buffer Manager	Total pages		48408	65792
7	SQLServer:Buffer Manager	Target pages		349566	65792
8	SQLServer:Buffer Manager	Database pages		7768	65792
9	SQLServer:Buffer Manager	Reserved pages		108	65792
10	SQLServer:Buffer Manager	Stolen pages		39465	65792
11	SQLServer:Buffer Manager	Lazy writes/sec		0	272696576
12	SQLServer:Buffer Manager	Readahead pages/sec		385	272696576
13	SQLServer:Buffer Manager	Page reads/sec		5625	272696576
14	SQLServer:Buffer Manager	Page writes/sec		2951	272696576
15	SQLServer:Buffer Manager	Checkpoint pages/sec		1686	272696576
16	SQLServer:Buffer Manager	AWE lookup maps/sec		0	272696576

Query executed successfully. | localhost (10.0 CTP) | JRMSERVER\Administrator (59) | Delldb | 00:00:00 | 1971 rows

Ready Ln 1 Col 45 Ch 45 OVR

Figure 12. Sample results from the following query: 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 Books Online: http://msdn2.microsoft.com/en-us/library/ms188754(SQL.100).aspx

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

Performance Data Collector and Management Data Warehouse

SQL Server 2008 x64 introduces the Performance Data Collector, whose goal is to maintain performance-related data in a format that helps administrators more easily tune their databases. The Performance Data Collector builds on SQL Server 2008 x64's Data Collector Architecture. (See msdn2.microsoft.com/en-us/library/bb677355(SQL.100).aspx for a discussion of the Data Collector Architecture.)

The Performance Data Collector is part of a broader set of capabilities that Microsoft refers to as the Performance Studio. The Performance Studio was not part of the November CTP, though it is supposed to be part of the upcoming SQL Server 2008 Readiness Kit. The Performance Studio allows you to collect performance data from multiple databases and store that data in a central repository. You can then compare current and past SQL Server performance.

The Performance Data Collector is the framework for collecting diagnostic and performance data. The Performance Data Collector stores the data it gathers in a special database Microsoft calls the Management Data Warehouse.

The Performance Data Collector uses three System Data Collection sets, which the SQL Server 2008 x64 documentation defines as follows:

- **Disk Usage.** Collects data about disk and log usage for all the databases on the server.
- **Server Activity.** Collects resource usage statistics and performance data from the server and SQL Server.
- **Query Statistics.** Collects query statistics, individual query text, query plans, and specific queries.

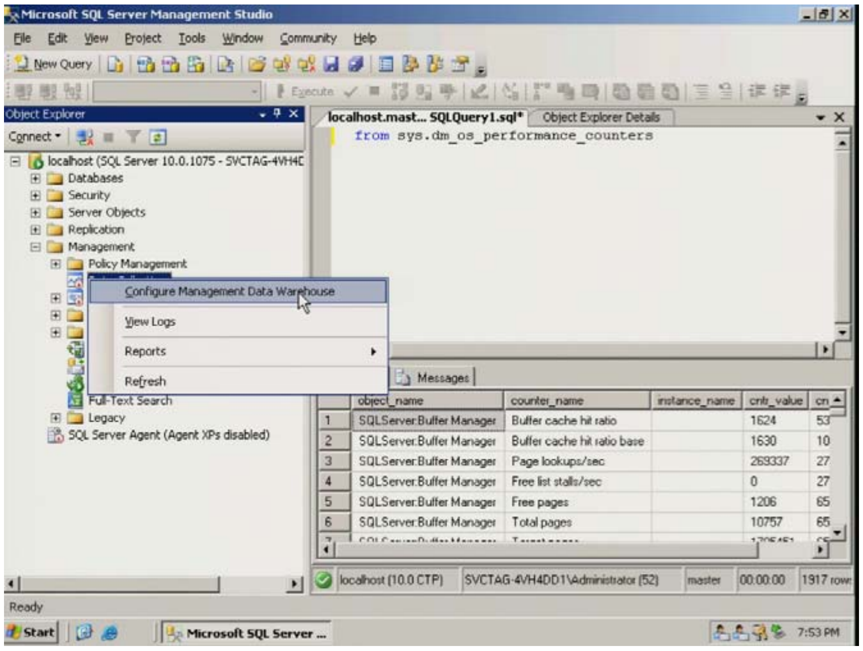
For more details about the specific data in each set, see http://technet.microsoft.com/en-us/library/bb964725(SQL.100).aspx

Once the Performance Data Collector is running, you can either wait for the collection schedule to cycle completely or update the data immediately. Once you have your data, you can view a historical report or you can create a custom report using any of the 20 fields available for each collection. We explain below how to update immediately and how to display a sample report.

NOTE: You must start the SQL Agent service before you configure the Performance Data Collector. In addition, you should set the SQL Agent service to start automatically so future data collection jobs will run.

To configure the Performance Data Collector, follow these steps:

1. Open Microsoft SQL Server Management Studio.
2. Connect to the server.
3. In Object Explorer, expand Management.
4. Under Management, right-click Data Collection, and select Configure Management Data Warehouse.


Microsoft SQL Server Management Studio
File Edit View Project Tools Window Community Help
New Query
Execute
Object Explorer
Connect
localhost (SQL Server 10.0.1075 - SVCTAG-4VH4D
Databases
Security
Server Objects
Replication
Management
Policy Management
Configure Management Data Warehouse
View Logs
Reports
Refresh
Full-Text Search
Legacy
SQL Server Agent (Agent XPs disabled)
localhost.mast... SQLQuery1.sql*
Object Explorer Details
Messages
Ready
Start
Microsoft SQL Server ...
7:53 PM

```
from sys.dm_os_performance_counters
```

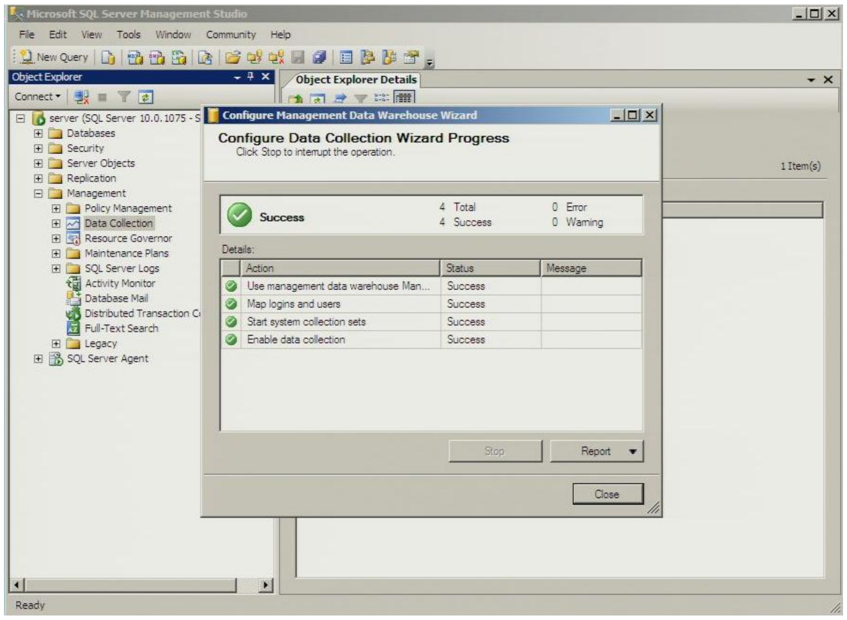
	object_name	counter_name	instance_name	cntr_value	cn
1	SQLServer:Buffer Manager	Buffer cache hit ratio		1624	53
2	SQLServer:Buffer Manager	Buffer cache hit ratio base		1630	10
3	SQLServer:Buffer Manager	Page lookups/sec		269337	27
4	SQLServer:Buffer Manager	Free list stalls/sec		0	27
5	SQLServer:Buffer Manager	Free pages		1206	65
6	SQLServer:Buffer Manager	Total pages		10757	65

localhost (10.0 CTP) | SVCTAG-4VH4DD1\Administrator (52) | master | 00:00:00 | 1917 rows

Figure 13. Configuring the Management Data Warehouse.

5. On the Welcome to the Configure Management Data Warehouse Wizard screen, click Next.
6. On the Configure Management Data Storage screen, select the local instance of SQL Server.
7. Click New, name the new database where you'll be storing the performance data, and click OK.
8. Select a file system location, which SQL Server uses to cache the performance data. This minimizes the impact of the data collection on the statistics.
9. Click Next.
10. On the Map Logins and Users screen, select the appropriate security, and click Next.

11. On the Complete the Wizard screen, click Finish.



Configure Data Collection Wizard Progress

Click Stop to interrupt the operation.

Success — 4 Total, 4 Success, 0 Error, 0 Warning

Details:

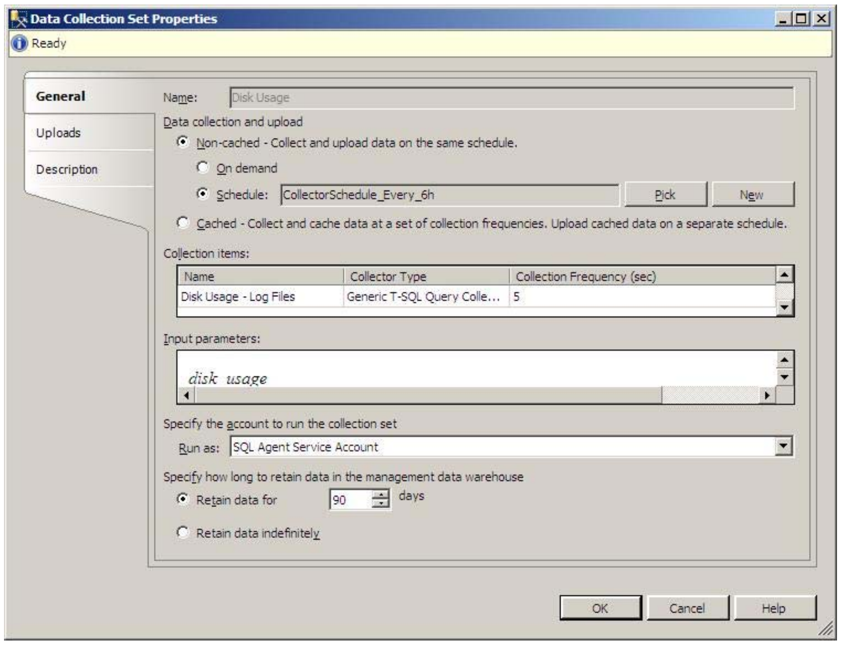
Action	Status	Message
Use management data warehouse Man...	Success	
Map logins and users	Success	
Start system collection sets	Success	
Enable data collection	Success	

Figure 14. Management data warehouse configuration completed.

The Performance Data Collector is now running.

You may configure each of the collection sets individually. To configure a collection set, follow these steps:

1. Open the Microsoft SQL Server Management Studio.
2. Connect to the server.
3. In the Object Explorer, expand Management.
4. Under Management, expand click Data Collection.
5. Right-click the collector you want to change, and select Properties.



Name: Disk Usage

Data collection and upload

- [✓] Non-cached - Collect and upload data on the same schedule.
  - On demand
  - [✓] Schedule: CollectorSchedule_Every_6h
- Cached - Collect and cache data at a set of collection frequencies. Upload cached data on a separate schedule.

Collection items:

Name	Collector Type	Collection Frequency (sec)
Disk Usage - Log Files	Generic T-SQL Query Colle...	5

Input parameters:

```
disk usage
```

Specify the account to run the collection set

Run as: SQL Agent Service Account

Specify how long to retain data in the management data warehouse

- [✓] Retain data for 90 days
- Retain data indefinitely

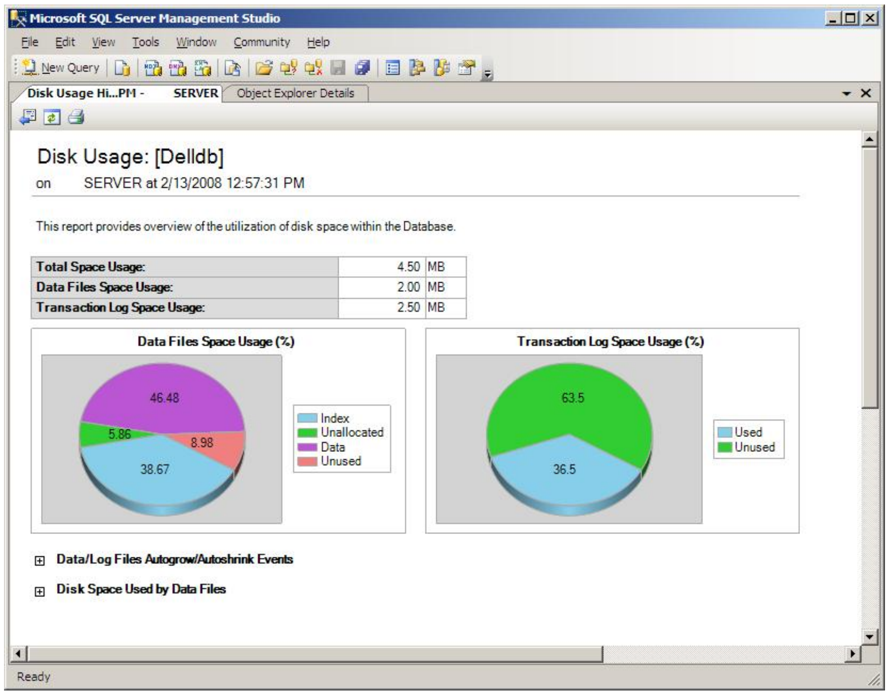
Figure 15. Disk Usage properties.

From the properties screen, you can set the schedule, period of data retention and other characteristics of the collection.

If you want to see a report before an entire reporting cycle has finished, you have the option of uploading the statistics early by following these steps:

1. Open the Microsoft SQL Server Management Studio.
2. Connect to the server.
3. In the Object Explorer, expand Management.
4. Under Management, expand click Data Collection.
5. Right-click the collector you want, and select Collect and upload now.

To view a report, follow the steps above, but select Reports, and select the report you want. The reports have hotlinks which allow you to drill down into the details of information of interest.


Microsoft SQL Server Management Studio
File Edit View Tools Window Community Help
New Query
Disk Usage Hi...PM - SERVER
Object Explorer Details
Ready

Disk Usage: [Delldb]

on SERVER at 2/13/2008 12:57:31 PM

This report provides overview of the utilization of disk space within the Database.

Total Space Usage:	4.50	MB
Data Files Space Usage:	2.00	MB
Transaction Log Space Usage:	2.50	MB

Data Files Space Usage (%)
46.48
5.86
8.98
38.67
Index
Unallocated
Data
Unused
Transaction Log Space Usage (%)
63.5
36.5
Used
Unused

⊞ Data/Log Files Autogrow/Autoshrink Events

⊞ Disk Space Used by Data Files

Figure 16. Sample management data report.

Summing up

Windows Server 2003 Enterprise x64 and SQL Server 2008 x64 introduce many new features and enhancements. As this Guide has documented, 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 2003 and SQL Server 2008 x64 provide a rich set of tools to help you do so. While a thorough treatment of monitoring a SQL Server 2008 x64 database is beyond the scope of this Guide, we introduced two tools you will find particularly useful for that purpose: the Windows Server 2003 Performance Monitor and the SQL Server 2008 Performance Data Collector. 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/windowsserver2003/default.mspx
www.microsoft.com/sql/2008/default.mspx

Appendix A. Methodology: Installing Windows Server 2003 R2 Enterprise x64

Install Microsoft Windows Server 2003 R2 Enterprise x64 by following these steps. We provide approximate times for each group of steps in the first step of that group.

1. Insert the Dell PowerEdge Installation and Server Management bootable CD version 5.2, and reboot the system. Note: During this process, the software will ask if you want to insert a replication floppy; we did not. (5 minutes, during which the system reboots twice.)
2. On the Welcome to the Dell OpenManage Server Assistant screen, click Click Here for Server Setup.
3. Set the date and time on the Set Date and Time screen. Click Continue.
4. On the Select an Operating System to Install screen, select Microsoft Windows Server 2003 Service Pack 1/Service Pack 2 x64 Edition. As noted above, we are installing Windows Server 2003 R2 Enterprise x64 – 5.2.3790 Service Pack 2 Build 3790. The following note appears on the screen: For Microsoft(R) Windows Server(R) 2003 versions, the R2 edition is also supported.
5. Leave Create Utility Partition at its default value of Yes, and click Continue.
6. On the screen Configure or Skip RAID, select Skip RAID configuration, and click Continue.
7. On the screen Configure the Physical Disk for Microsoft Windows Server 2003 Service Pack 1/Service Pack 2 x64 Edition, change the boot partition size to the maximum available – 69335 MB. Leave the File system at NTFS.
8. On the screen Confirm to Delete Partitions, click Continue.
9. On the screen Network Adapter(s) configuration, select Specify an IP Address.
10. Enter the IP addresses and select the correct subnet type.
11. On the screen Enter the Configuration Information for Microsoft Windows Server 2003 Service Pack 1/Service Pack 2 x64 Edition, enter your information. Be careful, after selecting Join Domain, to enter the domain name and password exactly. Otherwise, you will get errors later in the install process.
12. On the Operating System Installation Summary screen, click Continue. (6 minutes)

13. When prompted, insert the Microsoft Windows Server 2003 CD, so that Dell OpenManage Server Assistant can copy the operating system files. (5 minutes)
14. When prompted, remove the CD from the system, and click Finish. The installation begins. (27 minutes, during which the system reboots four times.)
15. When the system comes available, log in as Administrator. Note: the Administrator password is still blank at this point. When you log in, Windows asks for the second Windows server 2003 CD. Insert it, and click OK. (2 minutes)
16. Install SP2. We did this from a CD, accepting the license agreement and the default options. (3 minutes)
17. Reboot system. (4:35)
18. Install Windows update. We installed all essential updates. However, when prompted to install Internet Explorer 7, we declined. (Note: The time will vary; in our case, it was 10 minutes.)
19. Log in as Administrator. The password is blank at this point. Set the password for the Administrator account to a strong password.

Configuring data drives

Configure each of the two data (non-operating system) drives using the following process. These directions assume that no one has previously formatted the disks.

1. Click Start.
2. Expand Administrative Tools, and click Computer Management.
3. In the left panel, expand Storage, and select Disk Management.
4. Right-click the unformatted disk, and select New Partition.
5. Accept the default of Primary Partition
6. Accept the default Partition size in MB. This should be the entire disk.
7. Accept the default drive letter.
8. On the Format Partition screen, format with the File system set to NTFS and the Allocation unit sixe set to 64K. Filling in a Volume label is optional.
9. On the Completing the New Partition Wizard screen, click Finish.

Maximize throughput for network applications

We only needed to do the following steps for the first of the network connections. However, we did verify that the setting applied to the second network connection as well:

1. Click Start.
2. Expand Control panel, and click Network Connections.
3. Select Local Area Connection.
4. Select File and Printer Sharing for Microsoft Networks and click Properties.
5. On the Server Properties tab, select Maximize throughput for network applications.
6. Click OK.

Appendix B. Methodology: Installing SQL Server 2008 x64

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

We installed SQL Server 2008 using an ISO image we burned to a DVD. Microsoft's download page for the November 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 x64 by following these steps. We provide approximate times for each group of steps in the first step of that group.

Plan on 30 minutes to complete the installation.

1. Insert the SQL Server 2008 x64 DVD into DVD drive.
2. At the SQL Server 2008 x64 Start screen, under Install, click Server components, Tools, Books Online, and Samples. The windows disappears briefly (12 minutes)
3. On the Welcome to Microsoft .NET Framework 2.0 x64 Setup screen, click Next.
4. Accept the End User License Agreement and click Next. (56 minutes)
5. On the Setup Complete screen for .NET, click Finish.
6. On the Accept License Terms screen, check I accept the license terms, and click Next.
7. On the Installation Prerequisites screen, click Install to install the required prerequisites. (less than 1 minute)
8. On the Installation Center screen, click New Installation to launch the Installing SQL Server 2008 Wizard. The system configuration check runs automatically and should complete almost instantly.
9. Click Next to continue.
10. On the Feature Selection screen, select only those components you need for this particular installation of SQL Server 2008 x64. This practice reduces overhead on your Dell PowerEdge 2950. 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.

11. Click Next to continue.
12. 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.
13. Click Next to continue.
14. 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.
15. 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.
16. On the Account Provisioning tab of the Database Engine Configuration screen, select whether you want to run SQL Server 2008 x64 in mixed mode or Windows Authentication only. We recommend leveraging Active Directory security and choosing Windows Authentication. You must specify at least one SQL Server administrator. To add the account you are logged into Windows Server as an administrator, click Add Current User. If you run SQL Server 2008 x64 in mixed mode, you must enter the password for the “sa” account here.
17. 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.
18. Click Next to continue.

NOTE: To change the physical location of the user database files, user database log files, or tempdb directory files after the initial installation, use the ALTER DATABASE command from within the Microsoft SQL Server Management Studio. (Later in this appendix, we provide instructions for doing so.)

19. 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 wish to participate in the Microsoft data collection program.

20. Click Next to continue.
21. The Ready to install screen offers one last chance to review your choices. After doing so, click Install. (11 minutes)
22. At the confirmation screen, click Next.
23. At the Complete screen, note the Summary log file location, and click Close to complete the installation.
24. After the SQL Server 2008 x64 installation process completes, check the Microsoft Downloads Center (http://www.microsoft.com/downloads) for the latest SQL Server updates, patches, and service packs.
25. 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.
    a. Open SQL Server Management Studio.
    b. Enter the appropriate server name, enter credentials as necessary, and click Connect.
    c. Right-click Databases in Object Explorer, and select New Database.
    d. Enter the database name.
    e. Leave the owner as <default>.
    f. Scroll to the right of the Database Files window and enter the appropriate paths for the database file and log file. The default locations for these paths are those that you specified in the Data Directories tab of the Database Engine Configuration screen during the installation.
    g. 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 Delldb; named the data file Delldb.mdf; and called the log file Delldb_log.ldf. Be sure you have exclusive access to any database you plan to move.

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.

4. Right-click the database name whose file location you wish to change, and select Properties.
5. Under Select a page, select Files and take note of the logical name, path, and file name of the items you plan to move.
6. Click OK to close the Properties dialog. Note: System databases such as tempdb are visible in the System Databases view under the Databases view.
7. Right-click the database name whose file location you wish to change, and select Tasks | Take Offline. This step is unnecessary when moving tempdb files or tempdb logs.
8. When the Take Offline command completes, click Close to close the dialog box.
9. Use Windows Explorer to move the file or files to the new location. Note: SQL Server automatically re-creates the tempdb files and tempdb logs in the new location after the service restarts.
10. Under Databases | System databases, right-click Master, and select New Query.
11. For each file you moved, run the following statement:
    ```
    ALTER DATABASE database_name MODIFY FILE ( NAME =
    logical_name, FILENAME = ‘new_path\os_file_name’ )
    ```

    For example:
    ```
    ALTER DATABASE Delldb MODIFY FILE ( NAME = Delldb,
    FILENAME = ‘c:\newdbfolder\Delldb.mdf’ )
    ```
12. Right-click the database name whose file location you changed, and select Tasks | Bring Online. When moving tempdb files or logs, you must stop and restart SQL Server at this point.
13. When the Bring Online command completes, click Close to close the dialog box.
14. You can verify the file change by typing the following statements in a query window:
    ```
    Use database_name
    sp_helpfile
    ```

    For example:
    ```
    Use DellDB
    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 from the file system. This practice will free disk space and avoid potential 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.

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, respectively, of VeriTest.

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