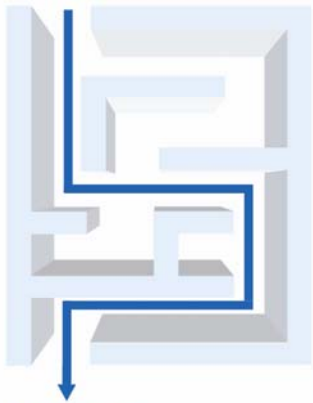


## DEPLOYMENT

---



GUIDE

SQL SERVER™ 2008 x64

ON

WINDOWS SERVER™ 2008 ENTERPRISE x64

ON

DELL POWEREDGE™ 2950

## Table of contents

---

Introduction .....	3
New features of SQL Server 2008 x64 .....	3
New features of Windows Server 2008 Enterprise.....	4
Installing and setting up Windows Server 2008 Enterprise x64 and SQL Server 2008 x64 in this environment .....	5
Defining our environment .....	5
Installing Windows Server 2008 Enterprise x64.....	6
Configuring the drives .....	8
Installing SQL Server 2008 x64 .....	11
Next steps: Best practices for administering and monitoring SQL Server 2008 x64 .....	16
Administering SQL Server 2008 x64 in this environment .....	16
<i>AUTOGROWTH</i> .....	16
<i>AUTO CLOSE</i> .....	17
<i>One tempdb data file per core</i> .....	17
<i>Processor and memory parameters</i> .....	18
Notes on the new SQL Server 2008 x64 compression capabilities .....	19
Monitoring SQL Server 2008 x64 in this environment.....	20
<i>Counters available from the Reliability and Performance         Monitor</i> .....	20
<i>Dynamic management views and functions</i> .....	20
<i>Performance Data Collector and Management Data         Warehouse</i> .....	22
Summing up .....	28
Appendix A. Methodology: Installing Windows Server 2008 Enterprise.....	29
<i>Set IP address, subnet mask, and domain information</i> ....	30
<i>Configuring data drives</i> .....	31
Appendix B. Methodology: Installing SQL Server 2008 x64	32
<i>Using the ALTER DATABASE command to modify the     location of user database or tempdb files and logs after an     installation</i> .....	34
About Principled Technologies.....	36

## 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 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 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.

## New features of Windows Server 2008 Enterprise

Windows Server 2008 Enterprise x64 also introduces many new features. As with SQL Server 2008 x64, there are far too many to include in this Guide. The following list mentions a few of the more significant features that may benefit database administrators:

- **Server core.** Windows Server 2008 Enterprise edition allows administrators to install only the functions necessary for a server's role. For example, the server core installation does not include a GUI or the Explorer shell. The reduced number of components should improve efficiency, reduce maintenance costs, and improve security by reducing the vectors available to viruses.
- **Windows PowerShell.** This new command-line shell includes over 130 utilities and an admin-focused scripting language to make system administration easier.
- **Self-healing NTFS file system.** If Windows® detects a file system error, it spawns a worker thread that attempts to fix this problem. If the problem is recoverable, the system will fix it without needing the administrator to restart it or shut it down.
- **Network Access Protection (NAP).** NAP allows administrators to define health and security policies for the network and to block non-compliant computers from the network.
- **Windows Deployment Services (WDS).** This replacement for Remote Installation Services (RIS) 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 on them.

- **BitLocker data encryption.** Windows Server 2008 Enterprise adds support for Microsoft's BitLocker™, which lets administrators easily and transparently encrypt whole volumes, even across multiple drives.
- **Built-in virtualization.** Windows Server 2008 Enterprise includes a hypervisor, Hyper-V™, which provides native virtualization. Hyper-V allows virtual devices to access the VMBus directly, via Enlightened I/O. This technology can greatly improve efficiency, but does require that the guest OS support Enlightened I/O.

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

---

In this section, we focus on installing and setting up Windows Server 2008 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](http://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.

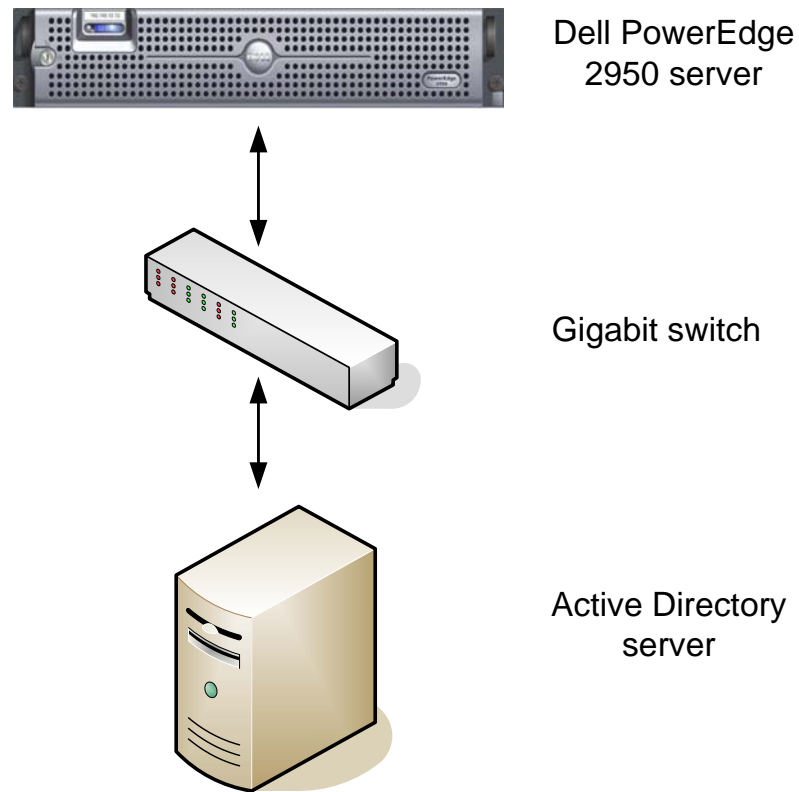


Figure 1. The setup we used in our hands-on testing and research for this Deployment Guide.

---

**★ 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.

The operating system on our server was Windows Server 2008 Enterprise x64, Service Pack 1. 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 available as of the start of our work on this Guide.

## 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 A provides full, detailed installation instructions.)

Because the Dell PowerEdge 2950 server's 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 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 ten minutes in our setup, will increase over time as Microsoft releases additional OS updates.

1. Boot the server using the Windows Server 2008 Enterprise x64 DVD. (2 minutes)
2. A screen appears prompting you to press any key to boot from DVD. Do so. In Windows Boot Manager, select Windows Setup, and press Enter. The installation software will begin to load files. (2 minutes)
3. Set the basic location information, enter your product key, select full install, and enter the information the installation software requires. Start the installation. (20 minutes, during which Windows resets the server twice)

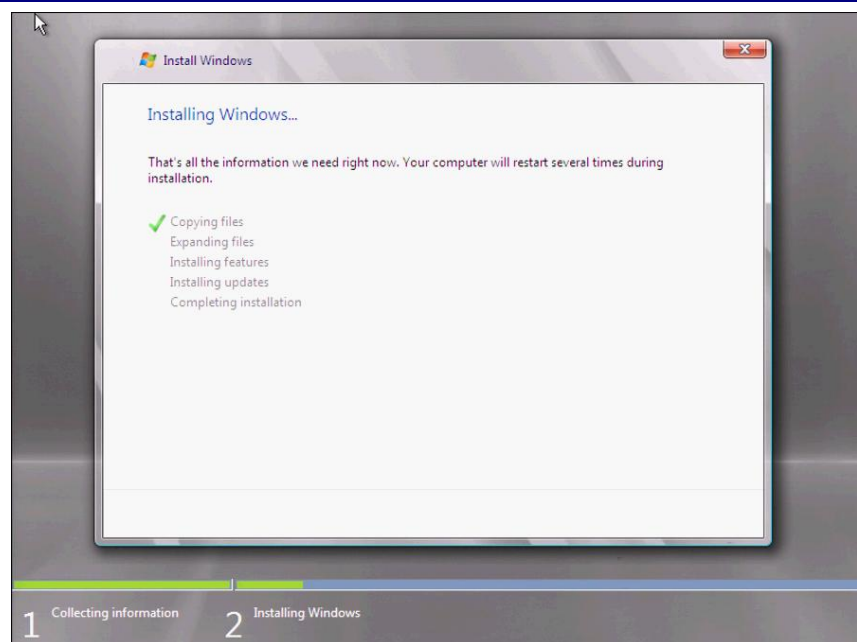


Figure 2. Windows Server 2008 installation progress.

4. Set your password; enter configuration information, including the IP addresses and domain information; and reboot the system. (2 minutes, 30 seconds)
5. Download and install updates. (The time will vary; in our case, it was 10 minutes.)
6. Reboot once the installation completes.

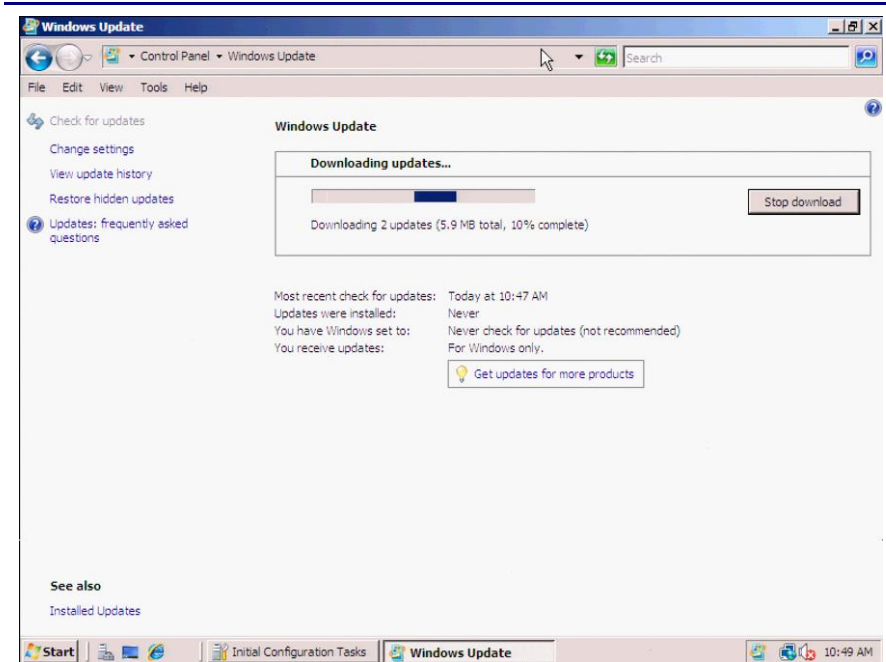


Figure 3. Installing Windows Server 2008 updates.

- ★ **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 2008 Enterprise 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: [http://msdn2.microsoft.com/en-us/library/ms143504\(SQL.100\).aspx#Review\\_NT\\_rights](http://msdn2.microsoft.com/en-us/library/ms143504(SQL.100).aspx#Review_NT_rights).

This section provides an overview of the SQL Server 2008 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.



**NOTE:** 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; accept the license agreement; and install the prerequisites. (1 minute)

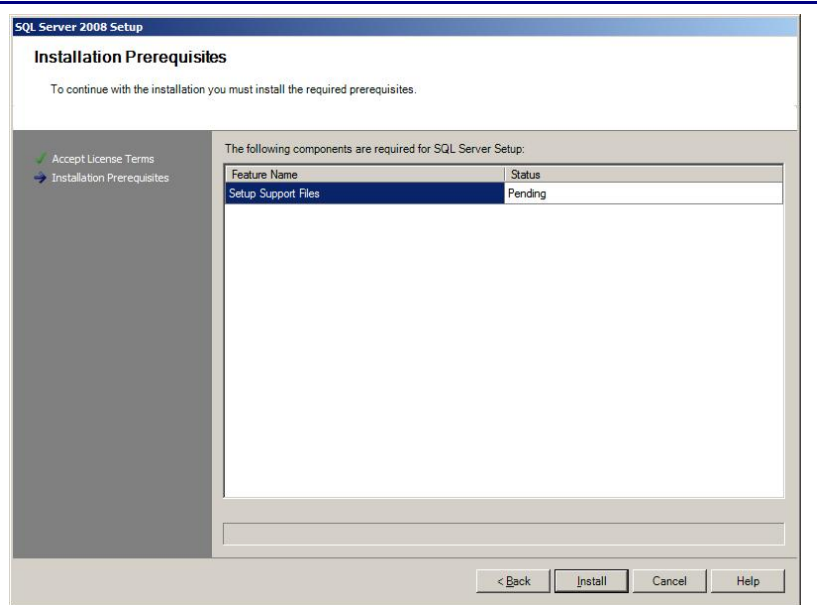


Figure 4. Installing SQL Server 2008 x64 prerequisites.

---

2. Click New Installation to launch the Installing SQL Server 2008 Wizard. The system configuration check completes almost instantly. Select the features you need to install. For a basic initial installation, install the Database Engine and Client Tools, as we did. You can install additional components later as necessary.
3. 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.

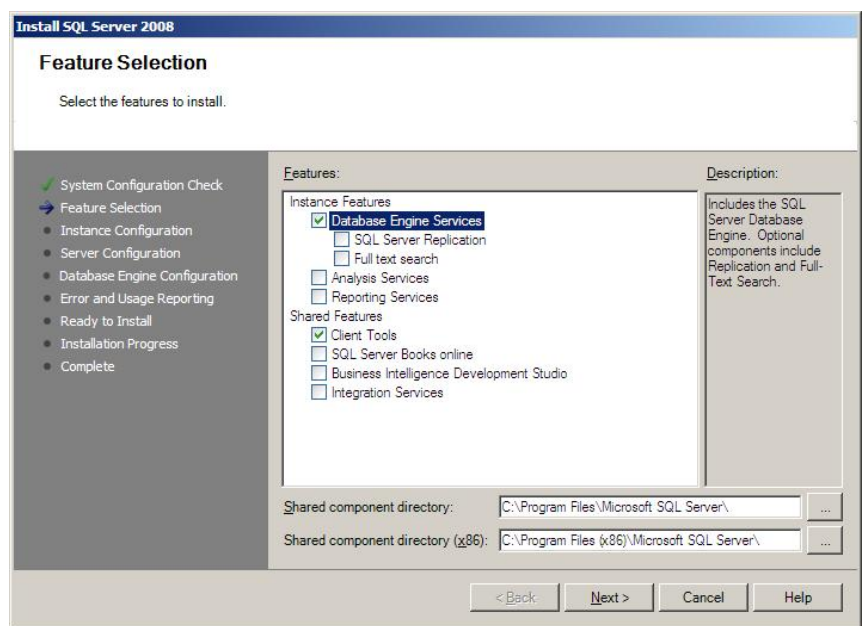


Figure 5. Selecting SQL Server 2008 x64 features.

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

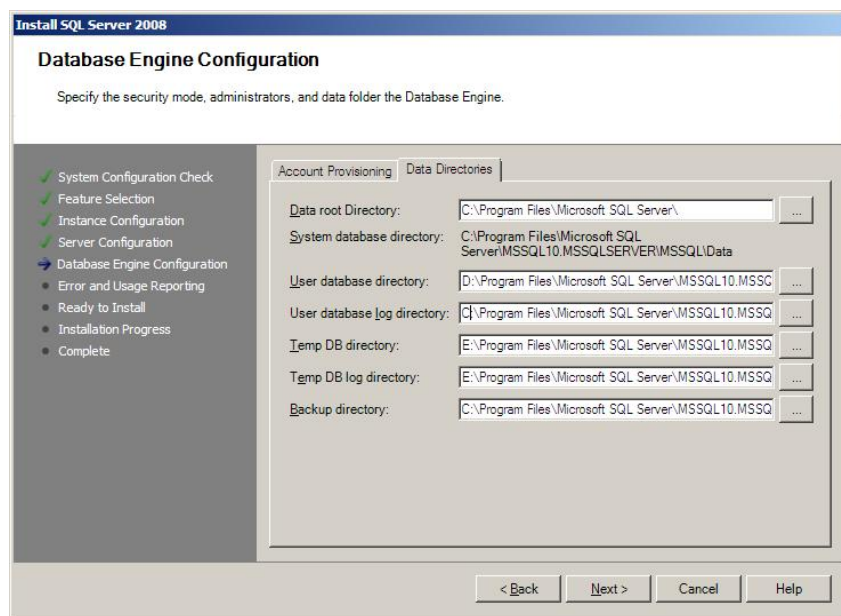



Figure 6. Setting file locations.

5. 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 2008 Enterprise provides a setting that allows SQL Server to retain its data in physical memory instead of swapping the data to virtual memory, the page file, or disk. To enable this setting, give the account running the SQL Server 2008 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.
5. Click Add User or Group, and enter the SQL Server service account name.

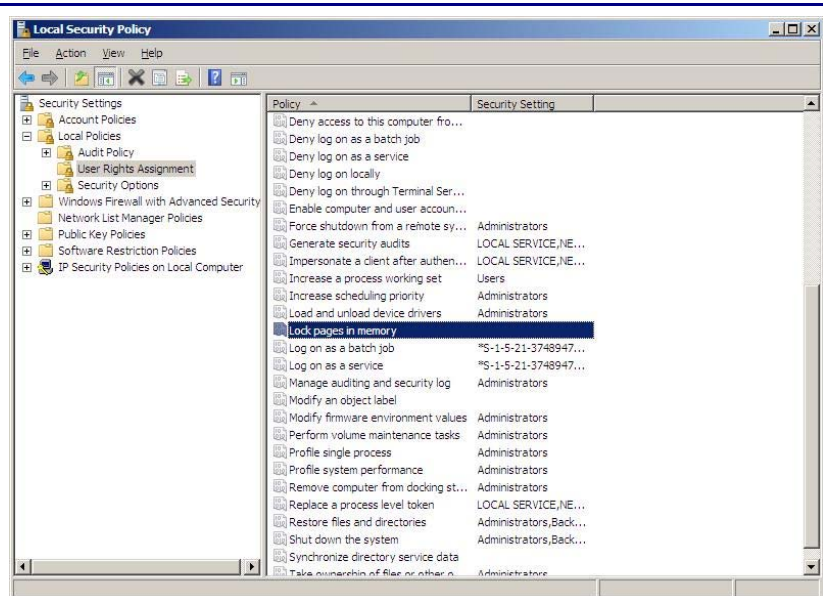


Figure 7. Selecting the Lock pages in memory setting.



**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 2008 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 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.

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 8. 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.

	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

Figure 9. 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](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](http://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](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.

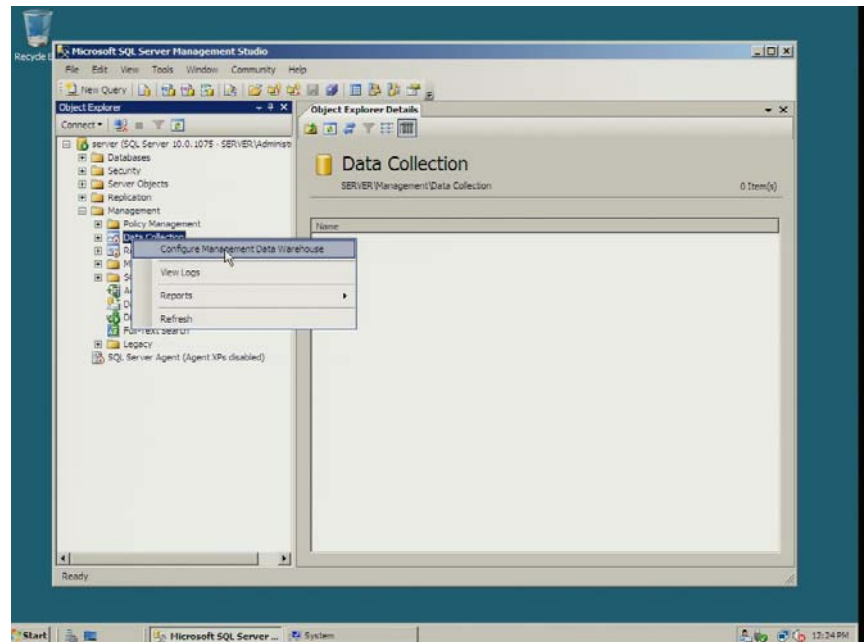


Figure 10. Configuring the Management Data Warehouse.

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.



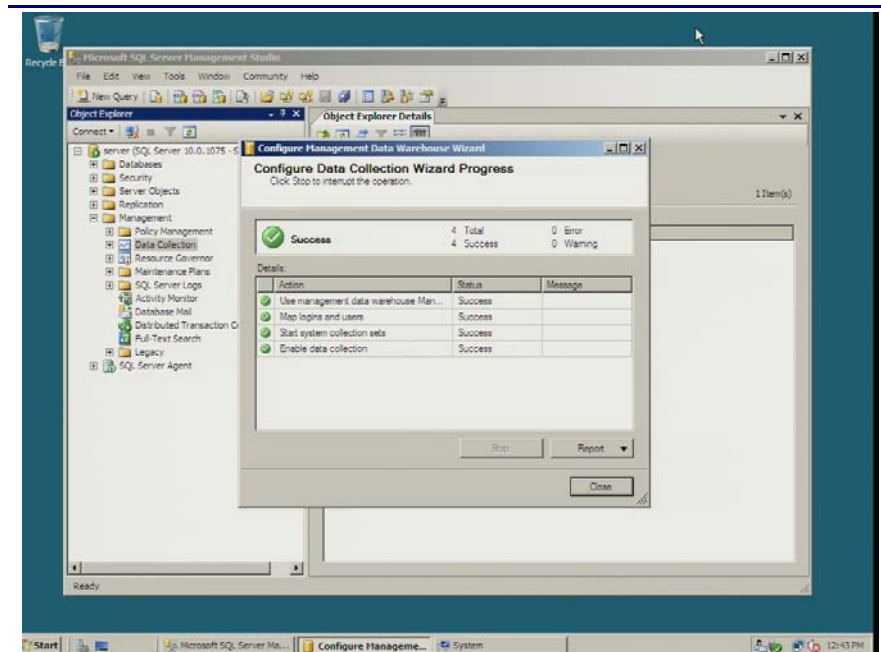


Figure 11. Management data warehouse configuration completed.

The Performance Data Collector is now running.

You can 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.

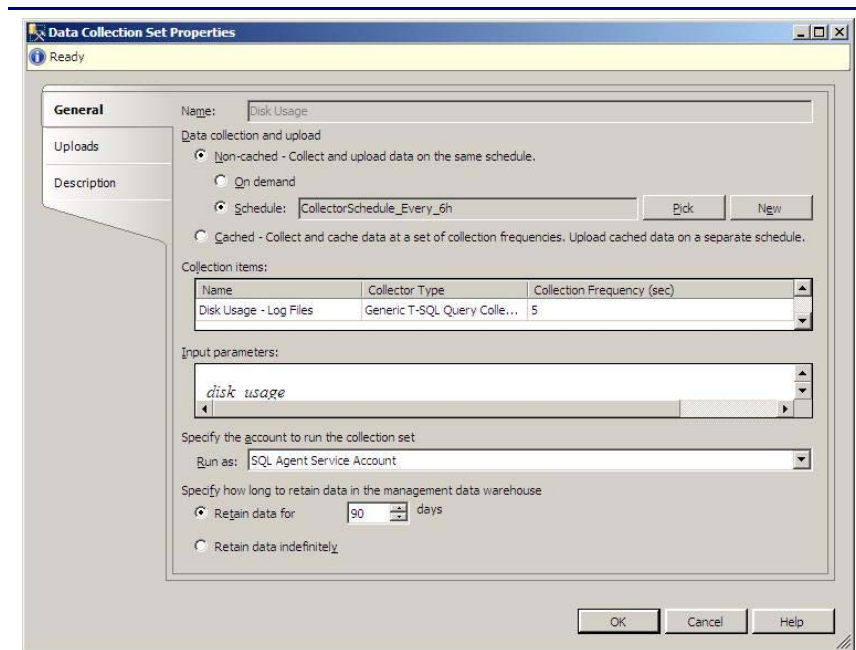


Figure 12. Disk Usage properties.

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.

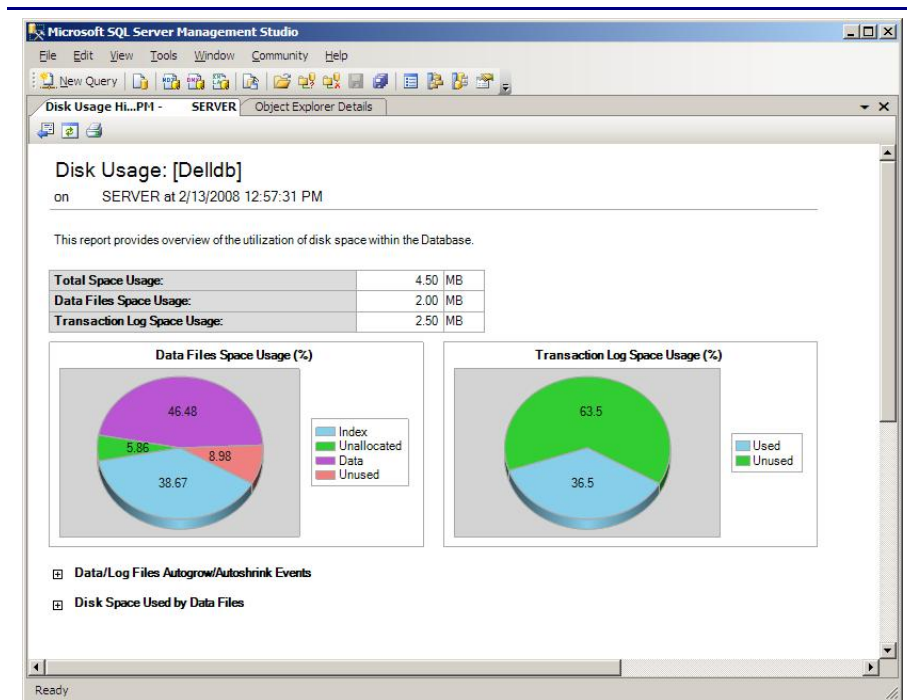


Figure 13. Sample management data report.

## Summing up

---

Windows Server 2008 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 2008 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 2008 Reliability and 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](http://www.dell.com/sql)

[www.microsoft.com/windowsserver2008/en/us/default.aspx](http://www.microsoft.com/windowsserver2008/en/us/default.aspx)

[www.microsoft.com/sql/2008/default.msp](http://www.microsoft.com/sql/2008/default.msp)

## Appendix A. 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. (2 minutes).
- 2.** When a screen appears prompting you to press any key to boot from the DVD, press any key.
- 3.** In Windows Boot Manager, select Windows Setup, and press Enter.
- 4.** During the boot, a Windows is loading files message appears, followed by a Windows booting loading bar. (2 minutes)
- 5.** 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.
- 6.** Click Install now.
- 7.** At the Type your product key for activation screen, enter your activation key, and click Next.
- 8.** In the pop-up window, click No.
- 9.** 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.
- 10.** At the Please read the license terms screen, check the I accept the license terms checkbox, and click Next.
- 11.** At the Which type of installation do you want? screen, for type of install, click Custom (Advanced).
- 12.** In the Where do you want to install Windows? screen, click Drive options (Advanced).
- 13.** Highlight any existing partitions, and click Delete.
- 14.** At the warning screen, click OK.
- 15.** When there are no more partitions, highlight the unallocated space, and click New.
- 16.** Click Apply.

- 17.** Highlight the new partition for the disk on which you plan to install the operating system, and click Format.
- 18.** Click OK.
- 19.** Highlight the partition and click Next to start the installation. (20 minutes, during which Windows restarts twice)
- 20.** When the system indicates that you must change the user's password before you log on the first time, click OK.
- 21.** When the installation process completes, click OK to change the password.
- 22.** Enter a strong password in both the New password and Confirm password fields, and click the arrow.
- 23.** At the Your password has been changed prompt, Click OK. Windows then prepares the desktop. (less than 1 minute)
- 24.** Install Windows updates. (Note: The time will vary; in our case, it was 10 minutes.)
- 25.** Reboot the server.

### **Set 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.

- 16.** When the Computer Name/Domain Changes window appears, click OK to close it.
- 17.** Click OK to go past the warning to restart.
- 18.** Click OK to close the System Properties window.
- 19.** Click Restart Now to reboot the server.

### **Configuring data drives**

Configure each of the two data (non-operating system) drives using the following process:

- 1.** Click Start.
- 2.** Expand Administrative Tools, and click Computer management.
- 3.** In the left panel, expand Storage, and select Disk Management.
- 4.** Right-click the unformatted disk, and select Format.
- 5.** Fill in a volume name if you want the disk to have one.
- 6.** Leave the File System as NTFS.
- 7.** Set the Allocation Unit size to 64 K. Windows Server 2008 will gray out the option to compress the drive.
- 8.** 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.

Install Microsoft SQL Server 2008 x64 Enterprise 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 Start screen, under Install, click Server components, Tools, Books Online, and Samples.
3. On the Accept License Terms screen, check I accept the license terms, and click Next.
4. On the Installation Prerequisites screen, click Install to install the required prerequisites. (less than 1 minute)
5. 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.
6. Click Next to continue.
7. 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.
8. Click Next to continue.
9. 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.
10. Click Next to continue.
11. 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.



12. 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.
13. 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.
14. 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.
15. 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.)

16. 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.
17. Click Next to continue.
18. The Ready to install screen offers one last chance to review your choices. After doing so, click Install. (11 minutes)
19. At the confirmation screen, click Next.
20. At the Complete screen, note the Summary log file location, and click Close to complete the installation.
21. 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.
22. 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.



Principled Technologies, Inc.  
1007 Slater Road, Suite 250  
Durham, NC, 27703  
[www.principledtechnologies.com](http://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, respectively, 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.

---