



EQUALLOGIC

TECHNICAL SUPPORT INSTRUCTIONAL SERIES

Technical Report	Document ID: TR1013
Deploying Microsoft® SQL Server In a Dell-EqualLogic PS Series iSCSI SAN	Revision: A03
	Revision Date: August 28, 2009

PS SERIES GROUPS DEPLOYING MICROSOFT® SQL SERVER IN AN ISCSI SAN

ABSTRACT

This Technical Report describes how to deploy SQL Server in an iSCSI SAN with PS Series storage arrays. It provides configuration and management recommendations for SQL Server and the PS Series storage.



Copyright © 2007 EqualLogic, Inc.

June 2007

EqualLogic is a registered trademark of EqualLogic, Inc.

All trademarks and registered trademarks mentioned herein are the property of their respective owners.

Possession, use, or copying of the documentation or the software described in this publication is authorized only under the license agreement specified herein.

EqualLogic, Inc. will not be held liable for technical or editorial errors or omissions contained herein. The information in this document is subject to change. Your performance can vary.

PS Series Firmware Version 3.2 or later.

EqualLogic, Inc.
110 Spit Brook Road
Building ZKO2
Nashua, NH 03062
Tel: 603.579.9762 Fax: 603.579.6910

TABLE OF CONTENTS

Table of Contents	iii
Revision Information	iv
Technical Resources	v
Introduction to SQL Server and PS Series Storage.....	1
Benefits of Deploying SQL Server with PS Series Storage Arrays.....	2
Planning and Design Considerations	3
Reliability, Availability, and Serviceability.....	3
Performance	3
Scalability	4
Recoverability	4
Ease of Management.....	4
Service Level Commitments.....	4
Optimizing Server Resources	6
Planning SQL Server Volume Configuration.....	6
SQL Server Volume Layout Guidelines	7
Sizing SQL Server Volumes	11
Distributing Data.....	12
Choosing Array RAID Levels	12
Deploying SQL Server with a PS Series SAN.....	13
Background Information.....	13
Basic Steps	13
Configuring SQL Server to Use PS Series Storage	14
Creating a User Application Database.....	14
Other Sources for Optimizing SQL Server	16
Migrating SQL Server Databases to PS Series Storage.....	17
SQL Server Data Management	18
Defragmenting SQL Server Files.....	18
Expanding SAN Storage Online	19
Data Protection with SQL Server	19
Backing Up Data Using VSS.....	20
Using PS Series Volume Replication with SQL Server	20
Summary	21
EqualLogic Documentation and Customer Support	21

REVISION INFORMATION

The following table describes the release history of this Technical Report.

Version	Date	Description
1.0	October, 2005	Initial Release
2.0	June, 2007	Added SQL 2005 coverage and support for PS Series 3.2 firmware release

The following table shows the software versions used for the preparation of this Technical Report.

Vendor	Software Component	Revision
Microsoft®	Windows® Server™ 2003	Service Pack 2
Microsoft	SQL Server 2005	Version 9.0.3042 Service Pack 2
Microsoft	SQL Server 2000	Version 8.00.2039 Service Pack 4
Microsoft	iSCSI Software Initiator	Version 2.04
QLogic	QLA405xC iSCSI HBA	BIOS: 1.09 FW: 02.0.0.45
QLogic	STORport Miniport Driver	2.1.3.28
EqualLogic®	PS Series Firmware	Version 3.2.0
EqualLogic	Host Integration Tools	Version 2.2.1

Technical Resources

The following table lists the documents and resources referred to in this Technical Report. All EqualLogic Technical Reports are available on the Customer Support site at:

<http://www.equallogic.com/resources/technicaldocumentsview.aspx?id=364>

Vendor	Document or Resource Title
EqualLogic	Deploying Windows Server 2003 in an iSCSI SAN Technical Report
EqualLogic	Microsoft Windows: Expanding Basic Disk Volumes Technical Report
EqualLogic	Storage Array Network Performance Guidelines Technical Report
EqualLogic	Boot from SAN with QLogic QLA405x HBAs Technical Report
EqualLogic	Deploying Thin Provisioning in a PS Series SAN Technical Report
EqualLogic	Microsoft Windows: Aligning Disk Sectors for Optimal Performance Technical Report
EqualLogic	Choosing a Member RAID Policy Technical Report
EqualLogic	Deploying Pools and Tiered Storage in a PS Series SAN
Microsoft	SQL Server 2005: Deployments and Tests in an iSCSI SAN
Microsoft	Recovery Models and Transaction Log Management
Microsoft	Microsoft SQL Server Best Practices Analyzer Tool
Microsoft	Capacity Planning for tempdb
Microsoft	Microsoft SQL Server 2000 - Deployment
Microsoft	Microsoft SQL Server 2005 – Deployment
Microsoft	How to move SQL Server databases to a new location by using Detach and Attach
Microsoft	How to Delay Loading of Specific Services
Microsoft	Microsoft SQL Server 2000 Index Defragmentation Best Practices
Microsoft	Best Practices Analyzer Tool for Microsoft SQL Server 2000 1.0
Microsoft	SQL Server 2005 Best Practices Analyzer (February 2007 CTP)
Microsoft	The SQL Server Health and History Tool (SQLH2)
Microsoft	Using the Copy Database Wizard
Microsoft	Large memory support is available in Windows Server 2003 and in Windows 2000
Microsoft	SQL Server 2005 Database Engine Backward Compatibility
Microsoft	Microsoft SQL Server 2005 Upgrade Advisor
Microsoft	Partitioned Tables and Indexes in SQL Server 2005

INTRODUCTION TO SQL SERVER AND PS SERIES STORAGE

As workloads grow, SQL Server databases continue to increase in size, requiring more and more storage capacity and performance. For example, when transaction rates increase, the storage subsystem must be able to handle the associated growth in disk space usage and I/O throughput. In addition, as database sizes increase, it becomes more difficult to complete backup operations in a timely manner. If an outage occurs, it can take many hours or even days to restore service.

SQL Server provides the enterprise data management platform your organization needs to adapt quickly to a fast-changing environment. Benchmarked for scalability, speed, and performance, SQL Server delivers rapid return on your data management investment, with low implementation and maintenance costs as well as rapid development of enterprise-class business applications.

A PS Series group—an iSCSI SAN consisting of one or more intelligent storage arrays connected to an IP network—is an excellent storage choice for SQL Server environments. PS Series storage provides fast setup, automated management, easy scalability, multipath I/O support, SAN boot support, and volume replication. Not only does a PS Series SAN improve storage utilization efficiency and availability, it also delivers flexibility and ease-of-management, regardless of SAN scale.

By utilizing the Microsoft Windows Server 2003 Volume Shadow Copy Service (VSS), the EqualLogic Auto-Snapshot Manager, and commonly available backup applications you can improve backup and restore operations through point-in-time copies of data called shadow copies or snapshots.

Thus, by deploying SQL Server with a PS Series SAN, businesses can combine the industry-leading database application with reliable, scalable, and high-performance disk storage to meet the ever-expanding needs of users.

To get the maximum benefits from SQL and a PS Series SAN, you should adhere to the best practices for SQL Server, as outlined by Microsoft and industry experts. In addition, this Technical Report describes requirements and recommendations for deploying SQL Server with PS Series storage arrays, including best practices for performance, reliability, scalability, flexibility, and recoverability.

Key issues in this report include the following:

- Design considerations, including availability, performance, scalability, and management.
- How to set up a PS Series group and volumes.
- How to set up a SQL server, including optimizing the server and connecting to volumes.
- Configuring SQL Server to use PS Series volumes.
- Expanding SAN capacity and file systems without affecting SQL operations.
- Backup and restore of SQL Server data.

BENEFITS OF DEPLOYING SQL SERVER WITH PS SERIES STORAGE ARRAYS

The benefits of deploying SQL Server with a PS Series SAN are as follows:

- **Rapid deployment and configuration of storage** – A simple setup utility (Remote Setup Wizard) lets you quickly configure an array on the network and create a PS Series group. Automation of complex operations like RAID configuration, disk sparing, data provisioning, and load balancing means that even novices can effectively manage the SAN.
- **Redundant hardware and hot serviceable configuration** – PS Series storage arrays are fully redundant with dual controllers, power supplies, and fans—all of which can be serviced online and without disrupting applications.
- **Data protection** – All data is protected with both RAID and spare disks. Combined with “hot” service capabilities, online operation is assured.
- **Simple and immediate storage expansion** – Using modular PS Series storage arrays you can increase SAN storage capacity and performance online, without server or application disruption.
- **SAN boot capability** – With iSCSI host bus adapters (HBAs) you can offload iSCSI processing from the server CPU and provide the ability to install and boot the Windows operating system from a PS Series volume, increasing flexibility and data protection. Other benefits of SAN boot include the ability to replace failed server hardware without having to re-install or move data. SAN boot also centralizes storage management and extends data protection to the system disks, and allows you to backup and disaster protect the boot disk by using SAN-based snapshot and replication operations.
- **Network path protection and load balancing** – Using multiple NICs or iSCSI HBAs with the Microsoft iSCSI Software Initiator you can configure multipath I/O and increase the reliability and performance of SQL Server. Also known as MPIO, multipath I/O enables dynamic load balancing of iSCSI SAN traffic across redundant paths between SQL Server and the PS Series SAN.
- **Advanced Management features** – PS Series storage comes standard with a comprehensive set of features including:
 - Automatic load balancing
 - Virtual volume management
 - Space-efficient snapshots for instant backup and restore
 - Volume cloning for rapid server provisioning
 - Multipath I/O (MPIO) support
 - Cluster support
 - Auto-replication for a comprehensive disaster recovery solution
 - Storage pools for creating a “SAN within a SAN”
 - Member-by-member RAID level control
 - Collections (consistency groups) tying volumes together across arrays and pools

- **VSS-based backups** – The EqualLogic Host Integration Tools software includes a VSS provider, called Auto-Snapshot Manager for Windows, which interacts with the PS Series storage, Microsoft Windows Server 2003, the SQL Server VSS writer, and a VSS requestor backup application to dramatically improve backup operations by creating flexible, space-efficient, point-in-time copies of data called snapshots or shadow copies.
- **Volume replication** – With PS Series auto-replication capability, SQL Server data can be automatically transferred to remote data centers, protecting the data from disasters – ranging from the destruction of the volume to a complete site disaster – with no impact on data availability or performance.
- **Thin provisioning database volumes** – With PS Series thin provisioning capability, storage administrators have the option to fully allocate logical resources while allocating fewer physical resources with automatic, on-demand growth. This type of storage allocation enables storage growth as needed without taking systems down and disrupting users. Repeated data growth operations can be avoided resulting in improved utilization of physical storage resources.

PLANNING AND DESIGN CONSIDERATIONS

When designing a SQL Server environment, you should understand the challenges of optimizing:

- Reliability, Availability, and Serviceability
- Performance
- Scalability
- Recoverability
- Ease of management
- Service level commitments

It is critical to focus on these challenges before you begin the initial SQL installation. The EqualLogic Technical Report, [Deploying Microsoft Windows Server 2003 in an iSCSI SAN](#), describes how to design a robust server and network environment that uses PS Series storage to overcome the challenges IT managers face today. This robust server and network environment will provide a sound base for your SQL Server environment.

Reliability, Availability, and Serviceability

The environment must be robust, resilient, and easily repaired by hot swapping components while the system remains available. In a SAN configuration, the server, network, and storage arrays should all have a high level of fault tolerance to avoid costly downtime and allow scheduled maintenance to take place without service disruption. Special attention should be paid to fault isolation and overall operations while system components are being serviced.

You can easily build fault tolerance into your SQL Server storage environment by using PS Series storage arrays, which provide high availability and scalability, in addition to hot-swappable hardware and MPIO support.

Performance

The SQL Server environment must provide optimal performance and response time. Access to SQL data is crucial to day-to-day operations, and under-performing storage can have a significant effect on the overall performance of SQL Server and on your company's productivity.

A SAN built with PS Series storage automatically optimizes its own performance and continually adjusts to changing demands. You can expand the SAN infrastructure online, and performance improves as you add arrays automatically.

Scalability

As your workload grows and storage capacity needs increase, the environment must accommodate changes without affecting users. PS Series storage arrays are self-scaling, with I/O performance that increases as the number of arrays in the SAN increases. When a new array is added, its resources are integrated into the SAN and immediately available to the virtualized storage pool. In addition to its storage capacity, the new array's disk spindles, controller cache, and GB Ethernet ports provide for enhanced performance.

Administrators can isolate the SQL Server workload from other application workloads within the SAN by creating separate storage pools for SQL Server database-related volumes within the PS Series group. Isolating specific application workloads can increase manageability and availability of application volumes across the PS Series SAN.

Recoverability

It must be possible to quickly recover from situations that can cause data loss, ranging in scale from a corrupted or infected file to the loss of an entire data center due to hostile action or act of nature.

PS Series storage arrays deliver snapshots and auto-replication technology as standard features, enabling automatic backup and disaster recovery capabilities for your critical data. Designing your environment with separate volumes for SQL Server databases and transaction logs can improve backup and recovery performance.

Ease of Management

Setup and configuration, backup and recovery, and day-to-day administration should be handled easily and have minimal impact on operations and users.

PS Series storage automates complex tasks such as RAID configuration and provides both graphical and command line user interfaces for easy and intuitive storage management.

Service Level Commitments

Organizations often have varying service level requirements across applications. One application might be critical to business results, while another might require segregation of storage resources to meet application best practices or organizational needs.

PS Series storage allows for differing performance and availability levels through the choice of arrays and RAID levels, while the ability to store data on physically discrete pools allows PS Series storage to meet data segregation commitments.

Figure 1 shows SQL Server configured to operate in a high-availability environment for a medium size organization.

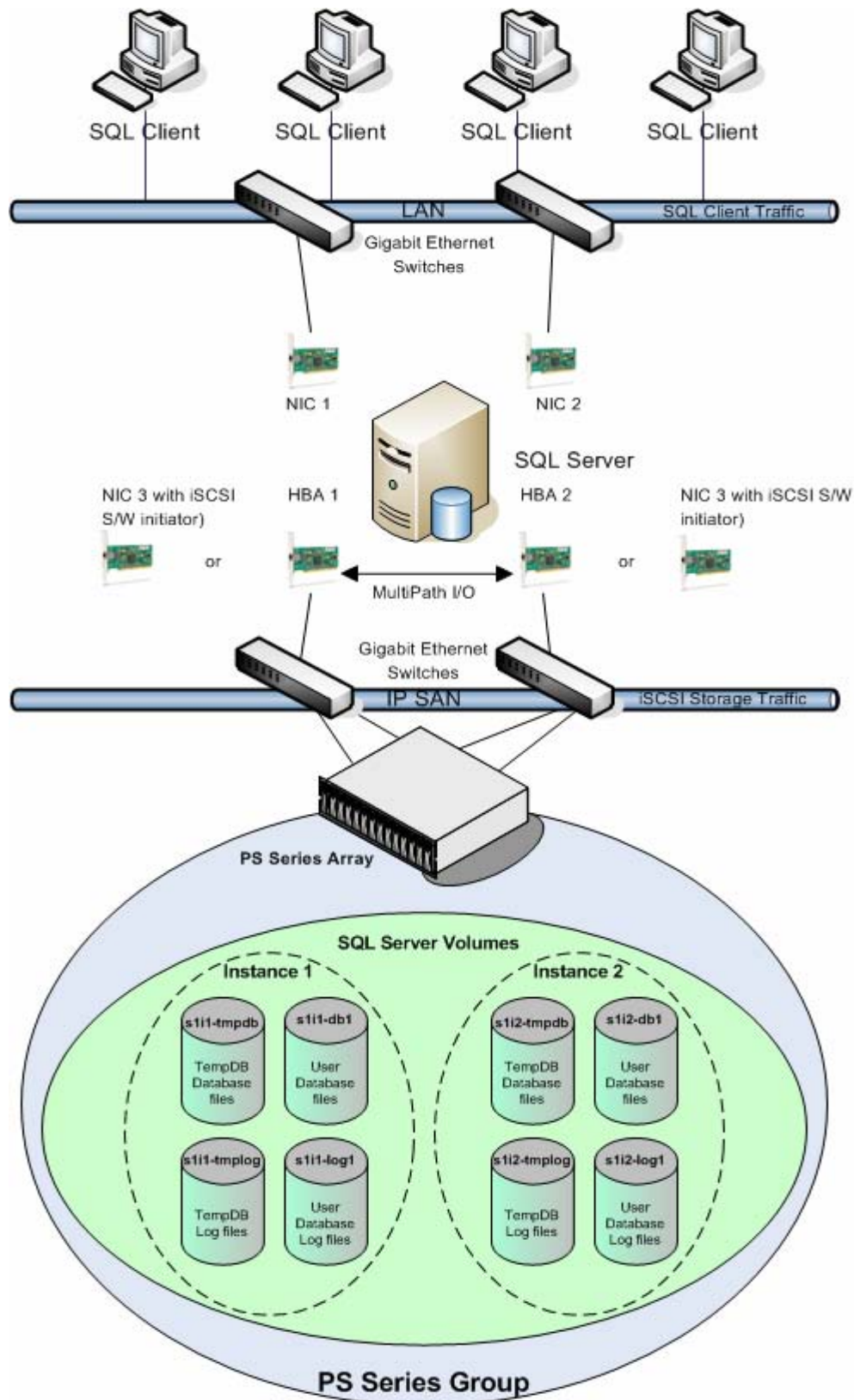


Figure 1: SQL Server High-Availability Design for a Medium-Sized Organization

Optimizing Server Resources

Prior to installing SQL Server onto an operating system it is important to make sure the operating system and system resources are sufficient to operate the SQL Server database environment you plan to run. That includes configuring your server with enough memory, processor, and network resources to accommodate your SQL Server deployment.

A good starting place is to follow the server optimizing recommendations in the EqualLogic Best Practice Technical Report [Deploying Microsoft Windows Server 2003 in an iSCSI SAN](#). It is also important to follow Microsoft's guidelines for minimum and recommended system requirements for deploying SQL Server.

Note: Microsoft's published minimum memory requirement for SQL Server 2000 Enterprise, Standard, and Developer Editions is 64MB, while 128MB or more is recommended. For SQL Server 2005 Enterprise, Standard, and Developer Edition, Microsoft requires 512 MB, while 1 GB or more is recommended.

It is important to properly configure your Windows 2003 server for optimal memory utilization. Memory configuration can affect the database I/O load and performance. A server configured with insufficient memory will generate heavier I/O loads to its databases, while configuring sufficient memory will make your SQL Server installation more efficient in its use of storage resources. Tuning your server resources appropriately will increase the overall efficiency of your environment, enabling you to gain the maximum benefit from your PS Series SAN.

The memory manager component of SQL Server 2000 eliminates the need for manual management of the memory available to SQL Server. When SQL Server starts, it dynamically determines how much memory to allocate, based on how much memory the operating system and other applications are currently using. As the load on the computer and SQL Server changes, the allocated memory also changes.

Some SQL Server 2005 memory enhancements include large memory support with the ability to enable and use Address Windowing Extensions (AWE) on the fly without restarting the SQL Server instance. AWE allows 32-bit applications to access physical memory greater than 4GB. (Note that this feature requires enabling the Lock Pages in Memory policy.) SQL Server 2005 also introduced the Memory Broker, whose primary role is to dynamically distribute memory between the buffer pool, the query executor, the query optimizer, and all the various caches. It determines distribution requirements through algorithms that analyze different types of workloads and calculate memory distribution across all components.

Planning SQL Server Volume Configuration

SQL Server data consists of numerous files with different functions. A SQL Server instance has multiple databases, and each SQL Server instance includes the following:

- System files (master.mdf, mastlog.ldf, model.mdf, modellog.ldf, msdbdata.mdf, msdblog.ldf, tempdb.mdf, and templog.ldf) hold the system information and temporary database and logs.
- Sample database files are created if the option to create sample databases is chosen at installation.
- Database files (*.mdf and *.ndf) are created by the user for specific databases that hold all the committed information.
- SQL transaction logs (*.ldf) are created for each database. The log files record all changes to a SQL database prior to the changes actually being committed.

SQL Server Volume Layout Guidelines

The goal when setting up SQL Server volumes in a PS Series group is to optimize manageability and performance according to your needs. You should consider how you will monitor disk usage and how you will perform backup operations. When creating volumes for use with SQL Server databases, best practices require aligning the disk sectors for volumes located on RAID sets. Misaligned volumes can directly impact storage RAID volume performance and defeat other performance optimization efforts. For more information on aligning storage volumes, see the EqualLogic Technical Report titled, [Microsoft Windows: Aligning Disk Sectors for Optimal Performance](#).

- **Database and log volumes.** When designing the volume layout scheme for SQL Server, there are many options. However, some general guidelines can be identified based on the size of your business.

By default, the SQL Server installation places the system and user databases and logs on the same volume. This design may be suitable for some organizations requiring very small and simple installations.

Microsoft recommends separating data and log files onto their own volumes in order to separate the random I/O going to the data files from the sequential I/O going to the log files. Before virtualization this was a common practice, but with today's data storage virtualization technologies it may have little or no impact on performance. (Note that in large enterprise database deployments this is still a performance best practice.) For smaller deployments the benefit is for database recovery. Having database and log files on separate volumes removes the risk of losing both files if the volume fails.

Note: Some applications require specific database volume layouts. Be sure to reference any application guides you may be using to determine if a best practice is recommended.

For small businesses needing a more robust SQL Server environment, a multi-volume deployment such as the one shown in Figure 2 is an alternative. Having separate volumes for each user database's data files and log files adds greater fault tolerance as well as ease of management. It is important to monitor the growth of tempdb and determine if it should also be placed on a separate volume.

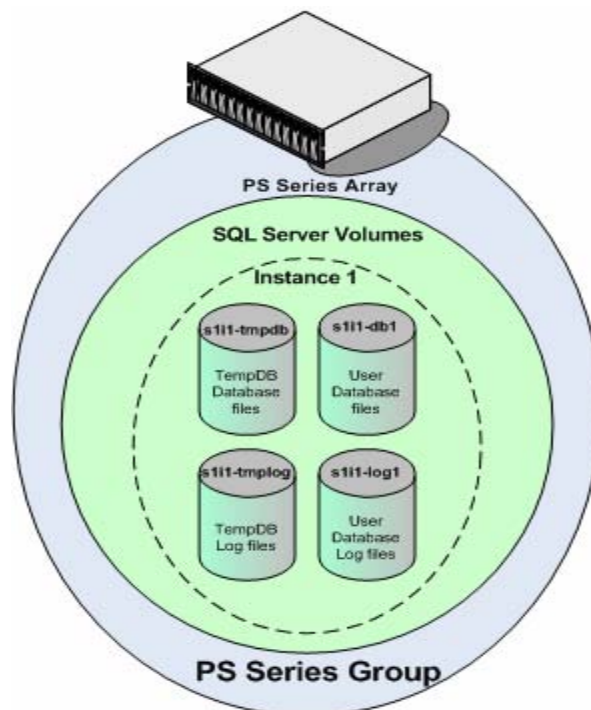


Figure 2: A Small Business, Multi-Volume Deployment.

A medium-size business that supports multiple user databases inside a single SQL Server instance can create a volume for data files, a separate volume for indexes, and another volume for log files, such as in Figure 3. This allows flexibility for database growth as well as isolating indexes and logs for performance and manageability. As with the small business deployment, it is important to monitor the growth of tempdb.

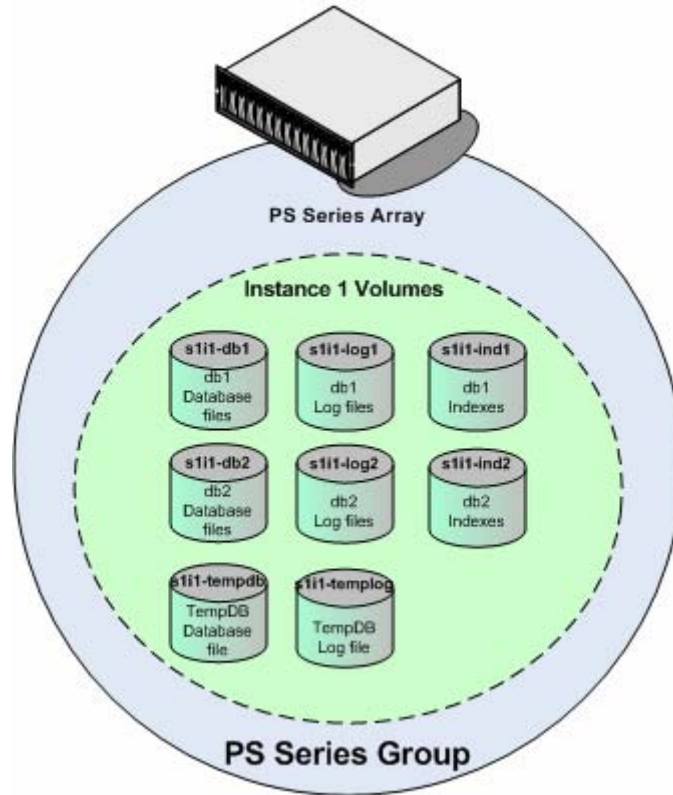


Figure 3: A Medium Business, Multi-Volume Deployment.

A large enterprise business running multiple databases and multiple SQL Server instances (Figure 4) may require a more complex volume layout. For example, you may need multiple volumes for user database, index, and log files as well as separate volumes to host the tempdb database.

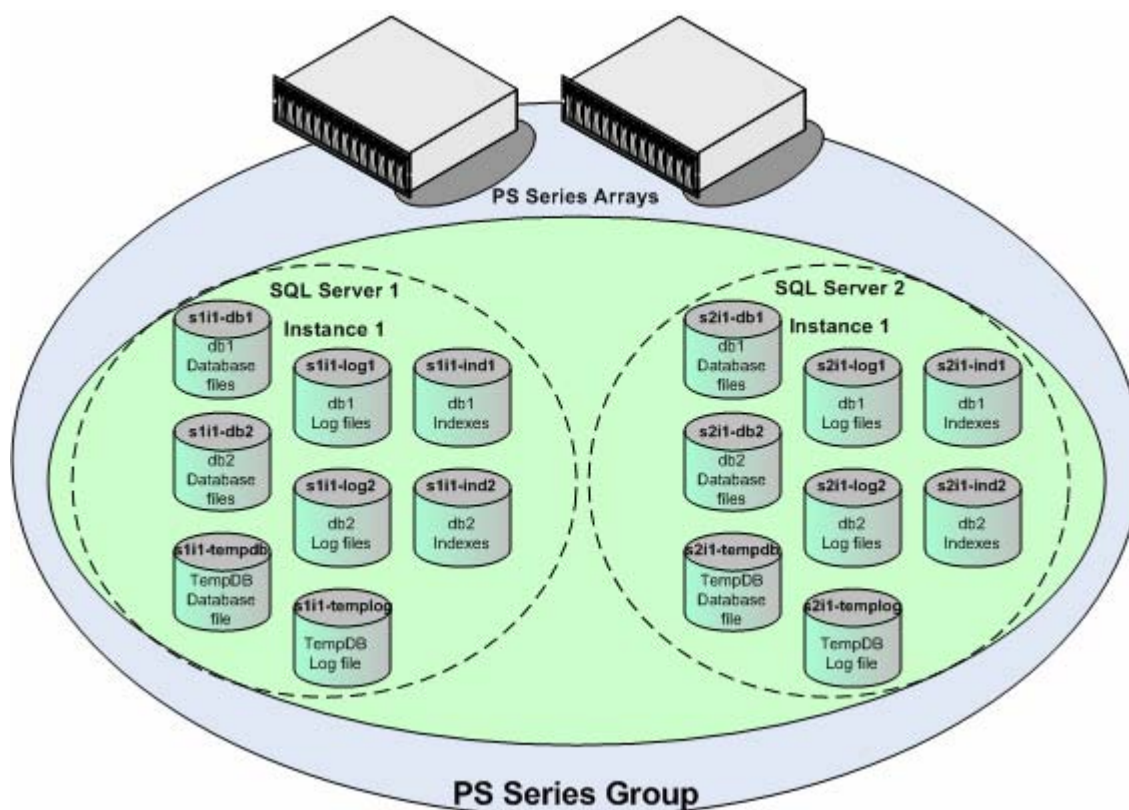


Figure 4: An Enterprise Business Deployment.

- **Database recovery model and volume layout.** The database recovery model you choose may determine whether or not to place log files on their own volumes. If your database is operating in the *Full recovery mode* where you are logging every transaction, then placing log files on their own volumes is a recommended to accommodate log file growth resulting from intense workloads. In this case, scheduled log backups are required to periodically truncate the log file. Databases running in Bulk-Logged or Simple recovery models do not account for every transaction and therefore do not require as much log space or growth planning.

Note: Changing the database recovery model will impact recoverability in the event of damaged or loss of data. For more information on database recovery models see the Microsoft article, [Recovery Models and Transaction Log Management](#):

[http://msdn2.microsoft.com/en-us/library/ms366344\(SQL.90\).aspx](http://msdn2.microsoft.com/en-us/library/ms366344(SQL.90).aspx)

- **Tiered database and log volumes.** PS Series SANs support tiered storage deployments. With SQL Server that can mean placing database and log files on specific volumes residing on the same or separate tiers. Tiers can be created using multiple storage types in the same or separate storage pools within a PS series group. By using a tiered design you can ensure I/O going to one pool is physically separated from I/O going to another pool. This can improve performance in some SQL Server environments. For more information on tiered storage designs see the EqualLogic Technical Report titled, [Deploying Pools and Tiered Storage in a PS Series SAN](#).

- **Create the database and log files as close to their estimated final size.** By default, SQL Server database file *autogrowth* is enabled when you create a new SQL Server database. Microsoft recommends using manual file growth options when possible to improve performance. By using PS Series storage, volumes can be increased online without incurring server downtime, although you must monitor space usage and administratively resize the volume. With thin-provisioned volumes, full storage requirements can be logically allocated while only physically using a fraction of the storage resources, and additional storage resources are allocated automatically. For more information on thin provisioning, see the EqualLogic Technical Report titled, [Deploying Thin Provisioning in a PS Series SAN](#).

You should try to obtain an accurate estimate of the final size of the database and log files for the following reasons:

- Every time SQL Server automatically grows a database or log file, it halts incoming transactions, consuming CPU and memory resources.
- Every time SQL Server must automatically increase transaction log size, more transaction log virtual files must be created and maintained by SQL Server. This increases recovery time if your transaction logs need to be restored.
- A *.mdf file will be fragmented only if you did not have enough contiguous space on the volume when it was created, or if it was resized at a later date and there was not enough contiguous space. Therefore, you should periodically defragment the file system, especially after you expand a SQL volume.
- **SQL Server tables and partitioning.** Partitioning can also help with table alignment and data management. In SQL Server 2005 table and index partitioning can greatly improve performance on large data sets.

In very large databases (VLDB), creating multiple file groups and placing each file group on its own storage volume can significantly improve performance and simplify management. By grouping or partitioning specific tables and indexes per file group rather than having them all on the same file group and PS volumes, you can take advantage of improvements in data parallelism in SQL Server 2005. When related tables and indexes reside on the same physical volume and are defined with the same partition function and partition scheme they become *storage-aligned*. The benefit of storage alignment in systems with multiple CPU's is that it allows processes to run in parallel without interruption or contention within the data scheme or physical storage.

Fragmentation can occur at the file table level. In SQL Server 2005 there is a new Database Management View (DMV), `sys.dm_db_index_physical_stats`. This new function allows you to determine the amount of index and table fragmentation. For more information on defragmenting SQL Server 2005 files see the section, **Defragmenting SQL Server Files**.

- **SQL Server database file groups.** When implementing multiple file groups per database it is recommended that the number of data files within a file group equal the number of processor cores in the server. It is also recommended to use compression for read-only file groups. This helps protect data and minimizes storage resource consumption. Any secondary file group marked as read-only can be placed on compressed NTFS volumes.

Note: In many cases I/O performance will improve when using compression for read-only file groups, but CPU performance may decrease due to the processing time required. You should place read-only file groups on compressed volumes in a test environment before implementing it in production.

- **Disable the automatic shrinking of databases.** By default, the SQL auto shrink option is disabled. If you allow SQL Server to automatically shrink its size, the space it de-allocates will leave fragments of unallocated disk space in the NTFS file system. Automatic shrinking of databases should be disabled to reduce fragmentation and overhead.

- **Tempdb database volume layout.** Tempdb is a database overlooked by many DBA's. In SQL Server 2005, tempdb plays a much larger role than in past versions. Tempdb has always been a version store where data is temporarily placed for read-only activity while another transaction is accessing the data within the database schema. SQL Server 2005 has increased the use of tempdb, and during long-running complex queries, index rebuilds, and similar operations. SQL Server will store the original data in tempdb until the query, rebuild, and update type operations are committed. This allows users read-only access to the original data while these new processes are taking place. As result, tempdb can become a potential performance bottleneck. It is important to understand the growth of tempdb over time and resize the tempdb database files accordingly – otherwise, every time SQL Server restarts, tempdb will shrink to its original size and the file growth process will start over. For more information on tempdb, see SQL Server Books Online and the following Microsoft article, [Capacity Planning for tempdb](http://msdn2.microsoft.com/en-us/library/ms345368(SQL.90).aspx).

[http://msdn2.microsoft.com/en-us/library/ms345368\(SQL.90\).aspx](http://msdn2.microsoft.com/en-us/library/ms345368(SQL.90).aspx)

- **Thin provisioning SQL Server volumes.** Thin provisioning allows the DBA or storage administrator to pre-allocate volume space but only physically use the space needed by the actual data on the volume. The advantage is that administrators can avoid having to periodically increase volume capacity on both the storage array and the SQL Server host. Another advantage is that storage will be allocated to the volume as it is needed, resulting in better storage utilization and less monitoring and growth management for administrators.

Thin provisioning is often used to manage storage for rapidly growing databases. The decision comes down to either plan on allocating a large physical storage volume to accommodate the database growth or enable pre-allocated thin provisioned volumes to grow on demand.

Thin provisioning is not appropriate for every deployment. For more information on thin provisioning see the EqualLogic Technical Report, [Deploying Thin Provisioning in a PS Series SAN](#) on the EqualLogic Customer website.

Sizing SQL Server Volumes

To size volumes for SQL Server databases, consider the space needed and the desired throughput. Determining the initial size of the database and log volumes depends on many factors, including how you plan to use SQL Server.

Considerations for sizing SQL Server volumes include the following:

- The workload being run on the SQL Server. An On Line Transactional Processing (OLTP) workload will generate very different I/O characteristics than On Line Analytical Processing (OLAP) or Decision Support System (DSS) workloads and should be sized differently.
- The database recovery model used. When using the Full database recovery model every transaction is accounted for by the database transaction log. The frequency of log file backup operations will determine the file growth rate and transaction log space you will need.
- The type of queries that will be performed each day. When running applications with long-running complex queries, you must optimize the tempdb system database. That might mean placing tempdb on its own PS Series SAN volumes.
- Whether your applications will create additional tables/indices inside the databases.
- Whether you expect to add databases in the future, and at what rate.

The best way to roughly determine the space required for a database volume is to base it on previous experience with similar databases or consult with the database administrator or application developer. You can also obtain a theoretical size that is based on the number of bytes in the fields and the number of rows.

The ability to increase volume size dynamically as your business grows and database usage increases is an important feature of a PS Series SAN. If you underestimate the size of a PS Series volume, you can quickly and easily increase its size. However, with proper planning, using this feature should be necessary only if unexpected changes to the environment occur, such as a merger with another company causing the database to grow substantially, or a new database experiencing unexpected initial growth.

Sizing for the optimal throughput may require you to deploy more volumes than what is needed to meet the database space requirements. For example, a 1-TB database will fit on one 1-TB volume. However, you may have better performance if you spread the database across more volumes, because more volumes can result in SQL issuing more I/Os in parallel – thereby improving throughput. Since a PS Series group dynamically load balances volumes by automatically spreading volume data across all the disks and arrays in the group, the more arrays in a group, the more disk drives are available for use.

Distributing Data

The load on a database file consists of random reads and writes. The load on a log file consists of sequential writes during normal operation and sequential reads during the restoration of a database. In traditional direct-attached (DAS) storage deployments, the write profile of these files can have a significant impact on performance because of the potential for I/O bottlenecks. However, the distinction in the write profile is less important when using a PS Series group because I/O is automatically distributed across arrays, RAIDsets and disks, dynamically balancing the load.

In a PS Series group, data distribution is handled automatically. The more arrays you have in the PS Series group, the more disks are available for the group to distribute data across.

Distributing database files and logs across file groups and volumes can provide more flexibility when backing up and restoring data. File groups allow for partial restore or ‘piecemeal’ database restore operations. Distributing data over multiple file groups and multiple volumes increases the availability of data sets.

Choosing Array RAID Levels

Before creating a PS Series group, you should determine which RAID level, RAID 10 or RAID 50, to configure for each individual group member (storage array). When choosing the RAID level for each array, you need to consider not only the performance characteristics of the various SQL Server files, but also the availability each level provides. Microsoft best practices call for deploying SQL Server log files on RAID 10 volumes where possible for best performance and protection from failures.

For large SQL implementations that require optimal performance, RAID 10 is a good choice for the group RAID level. When planning for IO intensive workloads such as OLTP, it is also recommended to place database log files on RAID 10 volumes. However, for many SQL implementations, RAID 50 can be used to provide maximum storage capacity, in addition to the performance benefits of striping.

Best Practice: RAID 50 is recommended for SQL Server volumes on PS Series storage to combine the advantages of performance and storage capacity.

In cases where you may want a mix of RAID 50 and RAID 10 storage volumes, a tiered storage design will be beneficial. You set up tiered storage by using different arrays with different RAID characteristics. This provides a more granular level of storage management as well as more control over IO designation to specific volumes (e.g., logs vs. database volumes).

A PS Series group’s automatic load balancing algorithms can determine RAID levels within a pool. Over time, volumes can be automatically moved to arrays offering the best RAID level for that volume. This activity happens automatically based on the performance needs of the volume. With a PS Series SAN an administrator can migrate a volume from one pool to another online. Administrators have the ability to move a volume from one storage array to another array by modifying volume settings.

For more information on RAID levels and storage tiering in a PS Series SAN, see the Technical Reports, [Choosing a Member RAID Policy and Deploying Pools and Tiered Storage in a PS Series SAN](#), on the EqualLogic Customer Support website.

DEPLOYING SQL SERVER WITH A PS SERIES SAN

The following sections describe SQL Server requirements and recommendations and the basic tasks for deploying SQL with a PS Series SAN.

Background Information

It is recommended that you are familiar with the following documents:

- Microsoft SQL Server 2005 – Deployment:
<http://www.microsoft.com/technet/prodtechnol/sql/2005/library/deployment.mspix>
- Microsoft SQL Server 2000 – Deployment:
<http://www.microsoft.com/technet/prodtechnol/sql/2000/deployment.mspix>
- Disk Sector Alignment:
<http://support.microsoft.com/default.aspx?scid=kb;en-us;929491>
- EqualLogic Technical Report, [Aligning Disk Sectors for Optimal Performance](#)
- EqualLogic Technical Report, [Deploying Microsoft Windows Server 2003 in an iSCSI SAN](#)
- EqualLogic Technical Report, [Deploying Pools and Tiered Storage in a PS Series SAN](#)

Basic Steps

Follow these steps to deploy SQL Server with PS Series storage:

1. **Set up a PS Series group and create the volumes required for the SQL environment.** See **Planning SQL Server Volume Configuration** for volume layout and sizing guidelines. Be sure to create access control records for each volume to allow the appropriate servers access to the volume. Also, reserve snapshot space for each volume if you will be creating snapshots or using Auto-Snapshot Manager for VSS operations.

See the PS Series *QuickStart* and *Group Administration* manuals and the EqualLogic Best Practice Technical Report [Deploying Microsoft Windows Server 2003 in an iSCSI SAN](#), located on the EqualLogic Customer Support website, for more information.

2. **Optimize the SAN network for performance.** See the EqualLogic Technical Report [Storage Array Network Performance Guidelines](#) for more information.
3. **Optionally, configure the environment so that SQL servers can boot from the SAN.** For details, see the EqualLogic Technical Report [Boot from SAN with OLogic 405xC HBAs](#).
4. **Prepare and optimize the SQL servers and connect them to volumes.** For details, see the EqualLogic Technical Report [Deploying Microsoft Windows Server 2003 in an iSCSI SAN](#).
5. **Install SQL on the server and configure it to use the PS Series storage.** See
6. *Configuring SQL Server to Use PS Series Storage* for information about specifying the iSCSI disks for the database and log files.

After deployment, you can expand iSCSI disks online, expand PS Series group capacity, and backup volumes, as described in *SQL Server Data Management*.

Configuring SQL Server to Use PS Series Storage

If this is an initial install or you are moving an existing SQL Server 2005 installation to use PS Series storage, there are some basic concepts to consider. It's important to create a volume naming scheme. This helps with management and understanding of what volumes are associated with specific applications as well as SQL Server database and log volumes. For example if you have a database named `MarketingDB` that spans two PS storage volumes, you may want to name the volumes associated with that database `SQLMKTDB-VOL1` and `SQLMKTDB-VOL2`. By keeping strict naming conventions, you create a much more manageable application storage environment.

In an existing SQL Server configuration, you can migrate or move your current SQL Server environment to a PS Series SAN by choosing one of the methods described in the section, **Migrating SQL Server Databases to a PS Series SAN**.

The following sections demonstrate using SQL Server with EqualLogic PS Series storage. Our example assumes a medium-size organization, with one user-created SQL database in one SQL instance on a single server.

Using the recommendations and guidelines outlined in **Deploying SQL Server with a PS Series SAN**, set up the SQL Server host to access the PS Series volumes. The example assumes a SQL Server 2005 instance is preinstalled on the local C: drive and all system databases are located on the local disk. The example describes installing a SQL Server database on PS Series volumes, and after the initial installation outlines how to move the tempdb database from the system database volumes to its own PS Series volumes. For this example we will use the drive letters and volume names below.

Drive	Volume Name	Description
C:\	Local	Program Files and System databases
R:\	user-db1.mdf	User database data
S:\	user-dblog.ldf	User database log
T:\	tempdb.mdf	Tempdb database data (post install)
U:\	templog.ldf	Tempdb database log (post install)

Creating a User Application Database

To create a new user database for your specific application, launch the SQL Server Management Studio, navigate to the desired SQL Server, right-click on **Databases**, and select **New Database** to open the **New Database** dialog box shown in Figure 5.

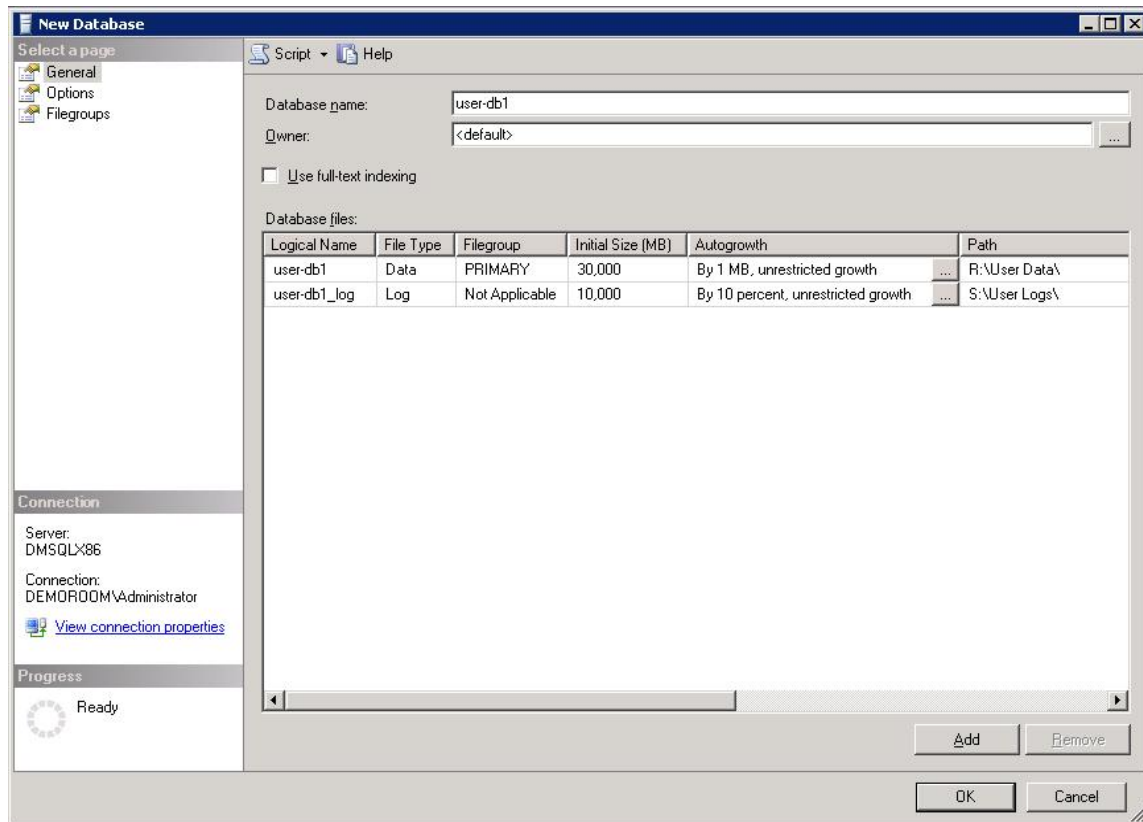


Figure 5: SQL Server Database Properties - Data Files

Enter a name for the new database, `user-db1` using the example configuration. Change the location of the database file to the iSCSI disk for this purpose (using the example configuration, the location would be iSCSI disk `R:` for data files and `S:` for log files) and specify the desired initial size of the database. Again, consult with your database administrator or application developer to determine the appropriate size.

Next, open a query window (Figure 6) to move `tempdb` files to the PS Series storage volumes. The move process will require restarting the SQL Server for the changes to take effect.

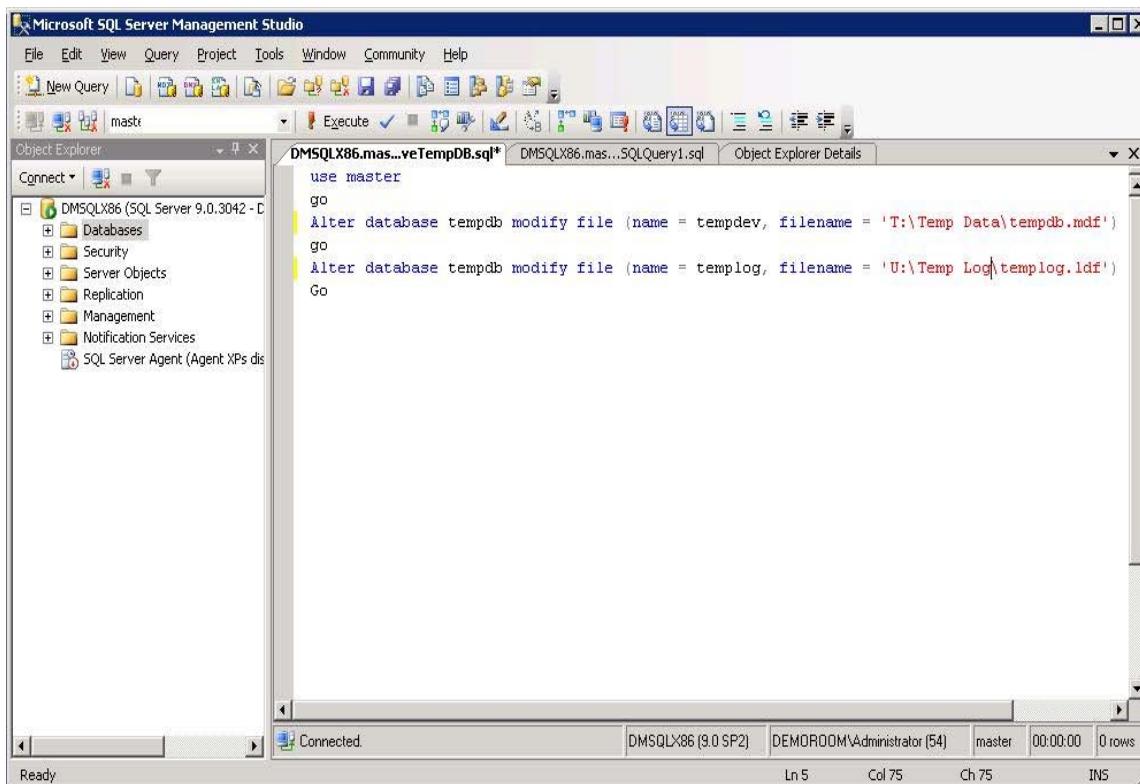


Figure 6: Move Tempdb Database

When the SQL Server database installation and setup is complete, using the example configuration, you should have the following database and log file paths configured:

Path	Description
C:\Program Files\Microsoft SQL Server\MSSQL.1	Program files
C:\Program Files\Microsoft SQL Server\MSSQL.1\MSSQL\Data	System databases
R:\User Data\user-dbl.mdf	User database data
S:\User Logs\user-dblog.ldf transaction logs	User database
T:\Temp Data\tempdb.mdf data	Tempdb database
U:\Temp Log\templog.ldf transaction logs	Tempdb database

OTHER SOURCES FOR OPTIMIZING SQL SERVER

There are many ways to tune and optimize your SQL Server environment. Visit the Microsoft SQL Server TechCenter for detailed information about SQL Server and links to information on past releases:

<http://www.microsoft.com/technet/prodtechnol/sql/default.msp>

SQL Server 2005 will make sure you meet the minimum hardware and software requirements to run SQL Server 2005 by running the System Configuration Checker during installation but a good way to find out if you can improve a current installation of SQL Server 2005 is to run the SQL Server Database Engine Tuning Advisor. This utility will detect if one or more settings need to be corrected. For more information, see the following Microsoft TechNet article:

<http://www.microsoft.com/technet/prodtechnol/sql/2005/sql2005dta.msp>

The Microsoft SQL Server Best Practices Analyzer Tool uses a predefined list of SQL Server recommendations and best practices to determine potential issues with your database environment. You can run the tool to detect if one or more settings need to be corrected or to ensure you SQL Server environment is operating correctly. For more information, see the following Microsoft TechNet articles:

Best Practices Analyzer Tool for Microsoft SQL Server 2000 1.0

<http://www.microsoft.com/downloads/details.aspx?displaylang=en&familyid=B352EB1F-D3CA-44EE-893E-9E07339C1F22&displaylang=en>

SQL Server 2005 Best Practices Analyzer (February 2007 CTP)

<http://www.microsoft.com/downloads/details.aspx?familyid=DA0531E4-E94C-4991-82FA-F0E3FBD05E63&displaylang=en>

The SQL Server Health and History Tool (SQLH2) allows you to collect and store information from your instances of SQL Server and then run reports to determine how SQL Server is being used. For more information on the SQL Server Health and History Tool, visit the following link to the Microsoft download.

<http://www.microsoft.com/downloads/details.aspx?familyid=EEDD10D6-75F7-4763-86DE-D2347B8B5F89&displaylang=en>

MIGRATING SQL SERVER DATABASES TO PS SERIES STORAGE

If you are currently using a local system disk, DAS, or other storage media for your current SQL configuration, you can easily migrate databases and logs to volumes in a PS Series group and gain the performance and scalability benefits of an iSCSI SAN. Several methods for migrating databases to your PS Series SAN are described below.

Method 1: Use Detach/Attach to move existing databases to a PS Series SAN.

Using detach and attach to move existing SQL Server databases is an easy way to migrate a database to a new location. Although this method is typically the fastest method to migrate a database, it does require scheduled database downtime and cannot be done with the database online.

As the user or DBA you can detach a SQL Server database using T-SQL or from the Management Studio GUI. Once the database is detached from the database server the database files can be copied to PS storage volumes and attached using T-SQL or the Management Studio GUI.

To move your existing user and system databases, refer to the Microsoft Knowledge Base article [How to move SQL Server databases to a new location by using Detach and Attach functions in SQL Server](#):

<http://support.microsoft.com/default.aspx?scid=kb:en-us:224071>

Method 2: Use Backup/Restore to move existing databases to a PS Series SAN.

This method of migrating existing databases to a PS Series SAN uses traditional methodologies for backing up and restoring SQL Server databases. Like the detach/attach method, this technique requires database downtime and could require extensive restore times depending on your environment. In this case you can backup an existing database as you normally would and then restore by pointing the database restore operation to the new volume mount points.

Method 3: Use the Copy Database option in SQL Server 2005 to move or copy existing databases to a PS Series SAN.

This method offers several options available through the Copy Database Wizard, including the ability to move the database while it remains online. The Copy Database Wizard gives you the choice of moving, copying, or upgrading an existing SQL Server database from one location to another. With the Copy Database Wizard you have the option to:

- Transfer a database when the database is still available to users.
- Transfer a database by the faster detach/attach method with the database unavailable during the transfer.

- Transfer databases between different instances of SQL Server 2000 or SQL Server 2005.
- Upgrade databases from SQL Server 7.0 to SQL Server 2000 to SQL Server 2005.

The Copy Database Wizard can perform all the tasks above, but there are restrictions on each. Note that the destination server must be running SQL Server 2005 SP2 or later. For more information on the [Copy Database Wizard](#) and options in SQL Server 2005, see the following Microsoft article.

<http://msdn2.microsoft.com/en-us/library/ms188664.aspx>

Note: To ensure optimal performance of an upgraded database, Microsoft recommends running `sp_updatestats` (update statistics) against the upgraded database on the SQL Server.

Table 1: SQL Server Database Migration

	DETACH/ATTACH MEHTOD	BACKUP/RESTORE METHOD	COPY DATABASE WIZARD METHOD
Online Migration	No	Only if restore to New Database	Yes (Side by Side)
Offline Migration	Yes	Yes	Yes (Attach/Detach)
Automated Operation	Yes (Wizard)	Yes (Wizard)	Yes (Wizard)
Manual Operation	Yes (Scripts)	Yes (Scripts)	No

SQL SERVER DATA MANAGEMENT

Data integrity is vital in a solid SQL Server environment. Backing up your data as well as monitoring for disk space usage are two important disk management tasks. You should monitor your SQL volumes so they do not fill up unexpectedly. When a volume starts to get full, you may want to consider increasing the volume size, as described below. Tested and documented backup and restore procedures are important in maintaining uptime of SQL Server environments.

Optionally, in SQL Server 2005 you can set the `PAGE_VERIFY_CHECKSUM` to increase data protection. The `PAGE_VERIFY_CHECKSUM` feature in SQL Server 2005 replaces the previous `TORN_PAGE_DETECTION` option in SQL Server 2000. When a page checksum error is detected, the DBA can restore the affected pages by using online page-level restore. This is a new feature in SQL Server 2005 Enterprise Edition.

Defragmenting SQL Server Files

Not only can the file system fragment, but SQL Server files and tables can also fragment over time, adversely affecting workload performance. Fragmentation can be avoided or at least reduced significantly by monitoring database volume growth and performing regular database maintenance tasks such as `DBCC DBREINDEX`, and recompiling stored procedures and triggers. If table rows are modified or deleted frequently, it is better to run intermittent `UPDATE STATISTICS` on the table(s) in question.

It is important to consider the workload type when determining whether or not to defragment. Microsoft reports that DSS workloads benefit much more than OLTP workloads due to the type of IO the workloads generate.

In SQL Server 2000, `DBCC SHOWCONTIG` allows you to determine the amount of fragmentation of index files. It is important to use necessary precautions before deciding to defragment SQL Server files. For more information, see the following *Microsoft SQL Server 2000 Index Defragmentation Best Practices*.

<http://www.microsoft.com/technet/prodtechnol/sql/2000/maintain/ss2kidbp.msp>

In SQL Server 2005 the DBCC SHOWCONTIG statement is replaced by the dynamic management function `sys.dm_db_index_physical_stats`. This new DMV will help detect fragmentation in a specific index or all indexes on a table or indexed view. With partitioned indexes, this DMV will also provide fragmentation information for each partition. You can run the DMV by the following syntax:

```
SELECT * FROM sys.dm_db_index_physical_stats  
  
DB_ID(N'<Database_Name>'), OBJECT_ID(N'<Table_Name>'), NULL, NULL ,  
'<SCANNING_MODE>');
```

The scanning modes available are LIMITED, SAMPLED, and DETAILED, with LIMITED being the default mode displaying page counts and external fragmentation without page density. The SAMPLED and DETAILED modes will analyze the entire table for both internal and external fragmentation.

To return information for all the tables and indexes, use:

```
SELECT * FROM sys.dm_db_index_physical_stats (NULL, NULL, NULL, NULL,  
NULL);
```

For more information on defragmenting SQL Server 2005 databases see SQL Server Books Online.

Expanding SAN Storage Online

As storage requirements grow for each application, PS Series storage can be easily expanded, online and with no disruption to users.

For example, you can increase the size of a PS Series volume using the Group Manager GUI or the command line interface (CLI). You must then enable the operating system to recognize the size increase. The additional space will be immediately available for use. This procedure is described in the EqualLogic Technical Report [Microsoft Windows: Expanding Basic Disk Volumes](#).

If the PS Series group does not have sufficient free space to increase a volume size or add new volumes, you will need to expand the group. To do this, simply add another array (member) to the group. See the PS Series *QuickStart* or the *Group Administration* manual for more information.

Data Protection with SQL Server

Data protection is crucial for database applications. Microsoft has embedded data protection and high availability capabilities into SQL Server. With past versions of SQL Server database administrators could backup and restore databases, cluster SQL Server instances and databases, replicate transactions across distributed servers, and perform protection scenarios such as Log Shipping.

With the introduction of SQL Server 2005, Microsoft has improved on these features and has added protection techniques such as Database Mirroring and Peer-To-Peer Transactional Replication. Database mirroring allows administrators to mirror database transactions either synchronously or asynchronously across the enterprise to a standby copy (mirror) of the database located locally or remotely. Peer-to-peer transactional replication allows distributed servers spread out across geographies to access the same data. Changes can be made by all nodes and the data is updated across all the distributed servers participating in the replication. If a server were to go down at one of the locations, the client application can be routed to another server in the replication model.

Backing Up Data Using VSS

You can back up SQL Server data residing on a PS Series storage array using any backup application that supports backing up SQL Server databases. However, to leverage the capabilities of Volume Shadow Copy Service (VSS) available in Windows Server 2003, SQL Server 2000 or 2005, and EqualLogic Auto-Snapshot Manager for Windows, you must choose a backup application that supports VSS, in addition to the backup features you desire. Note that backup and restore products are at various stages of VSS requestor implementation.

Some backup applications that support VSS and can be used as VSS requestors include:

- Symantec™ Backup Exec™
- CommVault® Galaxy™ Backup & Recovery
- CA BrightStor® ARCserve® Backup
- Bakbone® NetVault
- Legato® NetWorker

See the backup application documentation for information on using the application as a VSS requestor. See the *Auto-Snapshot Manager for Windows Installation and Administration* manual for information about using that VSS provider.

For more detailed backup and recovery procedures for specific backup and recovery vendors, see the Technical Reports on the EqualLogic Customer Support website.

Using PS Series Volume Replication with SQL Server

EqualLogic PS Series arrays include the ability to set up and configure PS Series volume replication between two PS Series storage groups, either in the same building or at a remote site for disaster recovery. PS Series replication technology is built into all PS Series array products as a standard feature. From a SQL Server standpoint, this means disaster recovery can be set up easily and efficiently.

PS Series replication requires two PS Series groups that are configured as replication partners. With replication, volume data can be automatically transferred across the groups, protecting volume data from failures ranging from destruction of the volume to a complete site disaster. Including all relevant volumes in a collection ensures that the database and log files are consistent with each other.

A typical disaster recovery set up will include a primary site that hosts all production data and a remote site to which data is replicated in case of a site failure. With replication, data protection can be set up for one way or multi-way replication, meaning that both sites host production data and replicate to each other.

A volume replica represents the contents of a volume at the point in time that the replication started. The first replica of a volume is a complete transfer of the volume data; all subsequent replicas consist of only the data that changed since the previous replica.

To recover SQL Server volume data after a disaster, the remote site PS Series group administrator can promote a replica set and bring up the SQL Server data on the remote site. Once the primary site is back up and operational, the group administrators can use the PS Series fail-back option and resync only the latest volume changes back to the primary site.

SUMMARY

An iSCSI SAN comprised of EqualLogic PS Series storage arrays provides an ideal storage infrastructure for SQL Server installations. A PS Series SAN brings all the reliability and performance needed for a successful deployment. As your SQL Server installation grows and the workload increases, a PS Series SAN can scale easily and dynamically while maintaining availability.

For a successful SQL Server installation, be sure to follow Microsoft's recommendations for Windows system and SQL Server configurations. In addition, you should follow the configuration best practices described in this Technical Report to ensure a robust installation that will meet your needs now and in the future.

EQUALLOGIC DOCUMENTATION AND CUSTOMER SUPPORT

Visit the EqualLogic Customer Service website, where you can download the latest documentation and firmware at <https://www.equallogic.com/login.aspx>. You can also view FAQs, the Knowledge Base, and Technical Reports and submit a service request.

EqualLogic PS Series storage array documentation includes the following:

- *Release Notes*. Provides the latest information about PS Series storage arrays and groups.
- *QuickStart*. Describes how to set up the hardware and start using a PS Series storage array.
- *Group Administration*. Describes how to use the Group Manager GUI to manage a PS Series group. This manual provides comprehensive information about product concepts and procedures.
- *CLI Reference*. Describes how to use the Group Manager command line interface to manage a group and individual arrays.
- *Hardware Maintenance*. Provides information on maintaining the PS Series storage array hardware.

To report a problem or to get access to the resources available on the Customer Support website, go to <https://www.equallogic.com/support/> and click LOGIN. If you do not already have an account, you can request one here. Once you are logged in, click Log a Case to report a problem.

To report an urgent issue, please call us at 1-877-887-7337 and speak with a member of the Customer Support team.

If you have any comments or suggestions related to this technical report, please send them to techreports@equallogic.com

Document Control and History Information

Revision	Change Author/Organization	Reason	Approved By/Organization	Date
A03	M. Cynewski / GCSS	ISO Compliance	M. Gauthier / GCSS	August 27, 2009
A02	Unknown	Added SQL 2005 coverage and support for PS Series 3.2 firmware release	Unknown	June 2009
A01	Unknown	Initial Release	Unknown	October 2005