



Best Practices

Best Practices for Installing and Configuring SQL Server 2005 on an LSI CTS2600 System

Table of Contents

_Toc269370599

Introduction	4
Configuring Volumes and File Systems	4
Configuring Volumes	5
Aligning Volumes	5
Formatting the File System	5
Pre-Installation Considerations	6
Setting Up SQL Server 2005 Accounts	6
Choosing What Components to Install	6
Choosing a System File Location	7
Choosing a Collation Setting	8
Choosing an Authentication Mode	8
Installing Service Packs and Hot Fixes	9
Configuring SQL Server 2005	9
Assigning a Location to the Database Files	9
Allocating Database File Space	11
Configuring Automatic Alerts with the SQL Server Agent	12
Configuring the Operators	12
Configuring the Alerts	12
Configuring the History	13
Configuring Auto Shrink	13
Backing Up and Restoring Files	13
Restoring the Database Owner	14
Scheduling Tasks That Result in Whole Procedure Cache Flushing	14
Using Standard Naming Conventions	15
Conclusion	16

Best Practices for Installing and Configuring

SQL Server 2005

Introduction

Database administrators today face many challenges as databases keep getting larger and data continues to be the basis used in making business decisions. Making sure that a database management system is installed and configured to meet the environment needs are top priorities for an organization.

This document focuses on installation and configuration best practices that will build a foundation for storage, performance, and assessment when using SQL Server 2005. This document does not replace the SQL Server 2005 installation instructions. However, it does give best practices at specific points. It should be read alongside the SQL Server 2005 installation instructions.

IMPORTANT Refer to Microsoft SQL Server 2005 documentation for performing the core step-by-step instructions for installing SQL Server 2005.

How to Use This Document

This document explains the best practices in the chronological order that they should be implemented during an installation. However, do not wait until you are installing SQL Server 2005 to read this document. Read this document before you schedule the installation, because there are several prerequisite configuration tasks. See "Configuring Volumes and File Systems" on page 2. Also, there are several choices offered during the installation that you must have considered in advance in order to know what choices to select. See "Pre-Installation Considerations" on page 3. When you configure the installed SQL Server 2005 instance, there are several parameters you will set, for which you must have fully considered the appropriate value. See "Configuring SQL Server 2005" on page 8. In addition, you should have already established standard naming conventions and communicated these with your IT staff. See "Using Standard Naming Conventions" on page 15. The best use of this document is to study it well in advance as an aid to planning the installation. Only after all the plans are in place, schedule the installation. Follow this document again during the scheduled installation to implement your plans appropriately. During the scheduled installation, use this document as a supplement to the SQL Server 2005 installation instructions.

Configuring Volumes and File Systems

There are several items you must configure to prepare the system for installation. Configure the system to make sure that segment size = sector size = cluster size (also known as allocation unit). In SQL Server 2005, all of these values should be set to 64KB.

Configuring Volumes

BEST PRACTICE: Configure the storage array segment size at 64KB to equal the SQL Server 2005 extent size.

When: Segment the volumes When: you create the hardware storage array and set up volumes at the storage array layer where the log files, data files, and tempdb files will be located.

How: Use the SANtricity Storage Manager software to configure storage array segment size. The opportunity to set the segment size occurs when you are setting up volumes at the storage array layer where log files, data files, and tempdb files will be located.

Benefits: Setting the proper segment size establishes correct alignment between the physical drive and the operating system file system. Correct alignment maximizes I/O request efficiency based on the SQL Server 2005 64KB extent space allocation value. SQL Server 2005 allocates space in 64KB extents.

Aligning Volumes

BEST PRACTICE: Sector-align volumes to 64 KB to equal the segment size of the hardware storage array.

When: Sector-align the volumes in the Windows operating system when you create the volumes and the partitions from the new storage arrays created in the previous step, "Configuring Volumes."

How: Use the DiskPart command line tool that is included with the Windows 2003 Server editions. In DiskPart, locate the partition that represents the volume you created

in the previous step, "Configuring Volumes." For more information, including the DiskPart tool commands and use, go to:

<http://www.microsoft.com>

Benefit: Aligning volumes maximizes the efficiency of I/O requests.

Formatting the File System

BEST PRACTICE: Format the Windows operating system file system using NTFS and a cluster size of 64KB to equal the sector size. Do not use the quick format option for formatting the Windows volumes.

When: Format the file system in the Windows operating system after you have aligned and created the volume and partitions in the previous step, "Aligning Volumes".

How: Use the NTFS file system with a 64KB cluster size when formatting Windows volumes where log files, data files, and tempdb database files will be located. Perform this formatting in the Windows OS, either by using the built-in GUI "Disk Management" or at the Windows OS command line with the format command. Matching the file system cluster size with the partition sector size provides maximum I/O request efficiency and maximum use of space between the SQL Server 2005 extents and the file system.

Benefits: The NTFS format process detects any bad sectors that might be on the Windows volume. The quick format option does not check for bad sectors on the Windows volume.

Pre-Installation Considerations

Several choices offered during the installation require advance consideration. Study the pros and cons of the upcoming choices ahead of time so that when you are installing you have already decided what you will choose. Then you will not have to interrupt the installation or because of time pressures make a poor choice that will have to be fixed later.

Setting Up SQL Server 2005 Accounts

BEST PRACTICE: Assign the least privileges possible to the local Windows user account and to the domain account used for SQL Server 2005 services.

When: Set up privileges for the SQL Server 2005 local Windows account and the domain service account during the installation.

How: During installation, the wizard prompts you to provide the service account information for SQL Server services to run under. SQL Server 2005 automatically adds to the new groups the same account that you specified during the installation to run the SQL Server 2005 service.

SQL Server 2005 automatically creates the necessary local Windows groups with only the least privileges needed for SQL Server 2005.

If the service account must change, you can add the new service account to the appropriate groups accordingly.

Benefits: Least privileges minimizes the risk of security-related exploits if the local Windows user account or the domain service account has been compromised.

Choosing What Components to Install

Additional components such as reporting services are not required but are offered as optional features. You might wonder whether adding optional components will be helpful or not. In fact, it is not helpful to add any component that is not necessary.

BEST PRACTICE: To use server resources efficiently, install only the components that are necessary.

When: During installation, the wizard allows you to select optional features.

How: Select only the following components. These components are all that are required for most environments:

- SQL Server 2005 Database Services
- Analysis Services
- Reporting Services
- Notification Services
- Integration Services
- Workstation Components

- Books Online
- Development tools

Benefits: Installing only the necessary components of SQL Server 2005 reduces space on the disk and reduces the overhead of operating system resources.

Each installed component requires disk space, and most components require operating system resources because they run as services in Windows. The amount depends on the component.

Choosing a System File Location

BEST PRACTICE: The system files installation target should be on a local, dedicated physical partition in either RAID1, RAID5, or RAID10 configurations. Do not share this partition with a physical partition where a memory paging file of the Windows operating system resides or with the root c:\ partition where the Windows operating system files are installed.

Install the system files on a local server storage target unless you are using SQL Server 2005 in a Windows cluster environment.

After the installation is complete, move the specific tempdb database data files and log files to their own dedicated physical partition for performance. (This task is not available in the installation routine.) Place the specific tempdb database data and the log files on either local or secondary storage.

When: You must choose a location for system files when you select which components to install.

How: When choosing which components to install, click the **Advanced** button for more options.

In the feature selection step of the installation, install the selected components (system files) on their own dedicated local (server) physical partition. Install the tempdb database on local storage or on secondary storage.

O **NOTE** If the tempdb database data and log files are on secondary storage and a failure occurs, you must configure SQL Server with a different file location path for the tempdb from the original when the SQL Server instance is restarted

Changing the locations *after* installation is possible but more complicated. Refer to the following additional documentation for the procedures:

- For information about changing database file location after the installation is completed, refer to Microsoft SQL Server 2005 documentation, specifically the DDL (data definition language) ALTER DATABASE command.
 - For steps on changing the tempdb database file locations and using SQL Server in single user mode during recovery processes after the installation is completed, go to <http://www.microsoft.com>
- Benefits:** Partitioning out the SQL Server 2005 system files prevents them from competing for disk resources with the Windows operating system such as a memory paging file.

A local (server) partition allows for flexibility in the event of secondary storage outages.

Configuring SQL Server 2005
The local system files/databases can still be accessed to restore user databases in the event of secondary storage outages.

Choosing a Collation Setting

The collation setting determines how the database will be sorted. The collation setting must be appropriate for the data.

BEST PRACTICE: Before selecting a default collation, research the application system requirements to make sure that sort order and case are compatible.

When: You choose a collation setting when selecting a collation during installation.

How: Choose the default collation setting during installation, using the installation wizard.

For more information about collation, see "SQL Server and Collation," by Muthusamy Anantha Kumar, *Database Journal*, January 28, 2004. Go to:

<http://www.databasejournal.com/features/mssql/article.php/3302341>
http://www.databasejournal.com/features/mssql/article.php/10894_3302341_2

Benefits: Choosing the correct SQL collation setting ensures compatibility with applications using SQL Server 2005.

Choosing an Authentication Mode

Because security is a high priority, you must choose the most appropriate authentication mode.

BEST PRACTICE: Use only Windows authentication mode when possible. Reserve the SQL Server 2005 authentication mode for special situations only.

When: During installation, the wizard prompts you to choose an authentication mode.

How: Consider the advantages and disadvantages of the authentication modes in advance so that you are sure which mode to choose when the installation wizard prompts you.

SQL Server 2005 supports two different authentication modes:

- **Windows** authentication mode is the most secure and flexible. Windows authentication offers flexibility in terms of managing logins for SQL Server 2005. Using Windows authentication mode exclusively is the most secure because it uses the Kerberos security protocol and the Windows policy features associated with logins.
- **SQL Server 2005** authentication mode (also known as mixed mode) lets you run both Windows authentication and SQL Server 2005 authentication at the same time. Mixed mode should be used only when the environment requires it. For example, use mixed mode for specific application connectivity requirements.

If you must use SQL Server 2005 authentication mode, and the sa account is not

being used, disable the sa account because the sa account is granted high privileges by default.

Benefits: Using Windows authentication mode exclusively is the most secure because it uses the Kerberos security protocol and the Windows policy features associated with logins. You can still switch to SQL Server 2005 authentication mode if your environment demands it.

Installing Service Packs and Hot Fixes

BEST PRACTICE: Always install the latest SQL Server 2005 service packs, but install hot fixes only if required and only after proving them in a test environment. Back up the production system database and the user's database first so that the databases can be restored if necessary.

When: Install SQL Server 2005 service packs immediately after base installation. Apply hot fixes as they become necessary and available. Apply hot fixes only after careful planning, as described in the following section.

How: Install the most current service pack release that can be found on the Microsoft web site. Research and evaluate whether each hot fix is necessary. Apply hot fixes *only* if the SQL Server 2005 system is experiencing a known, related issue.

After you determine that applying a hot fix will remedy a known issue, proceed with caution. First apply hot fixes in a test environment. If the trial is satisfactory, carefully plan how to best back up the production system database and the user database. Next, perform the backup as planned. Only after all this, apply the hot fixes to the production environment. Be prepared to remove them and restore the databases if necessary.

Benefits: Installing the most current service pack helps to make sure that the installation includes the latest security and product updates.

Applying hot fixes in a test environment first minimizes the risk of down time when they are applied to a production environment. Backing up the production system database and the users database helps to make sure that these databases can be returned to a point in time before applying hot fixes.

Configuring SQL Server 2005

Assigning a Location to the Database Files

BEST PRACTICE: Separate the following database files based on their numbers of of I/O requests:

- **SQL Server 2005 system databases**
- **tempdb**
- **SQL Server 2005 user databases**

When: Wait until after installation to separate the database files that must be separated.

How: To determine the most current database file location, query the file catalog view `sys.master_files`.

Configuring SQL Server 2005

SQL Server 2005 system databases (master, model, msdb, distribution), data files, and log files can be grouped together and can share the same dedicated volume (Windows disk) because these databases primarily store metadata.

The tempdb system database is I/O-intensive, thus providing many temporary object mechanisms for processing workloads. For this reason, the tempdb data and log files should have their own exclusive volume.

Change the location of the production system database and the user database by using the Data Definition Language (DDL) features in SQL Server 2005.

SQL Server 2005 user database data files typically have a random I/O request workload. Log files have a sequential I/O request workload. Put these different user database file types on dedicated volumes (Windows disks) to match the I/O request workload with the database file type.

To improve performance, create secondary index files in their own file group on those user databases that are earmarked for containing non-clustered index data structures. Place the file group and associated files on their own dedicated volume (Windows disk). This separation lets maintenance activities, such as rebuilding non-clustered indexes, occur separately from the primary data files.

Install SQL Server 2005 files on partitions that are part of a RAID configuration. The best RAID level depends on several variables within an organization such as performance, availability, and cost.

SQL Server 2005 databases have at least two files associated with them: a data file

(* .MDF), and a log file (* .LDF). You also can add additional secondary files (* .NDF).

For OLTP environments, install the data files, log files, and tempdb files on RAID10 configurations. For OLAP environments, install the data files and log files on RAID5 configurations and install the tempdb files on RAID10 configurations. For both OLTP and OLAP environments, install system data files on a RAID1 configuration or on a RAID5 configuration.

Figure 1 shows a typical database file distribution.

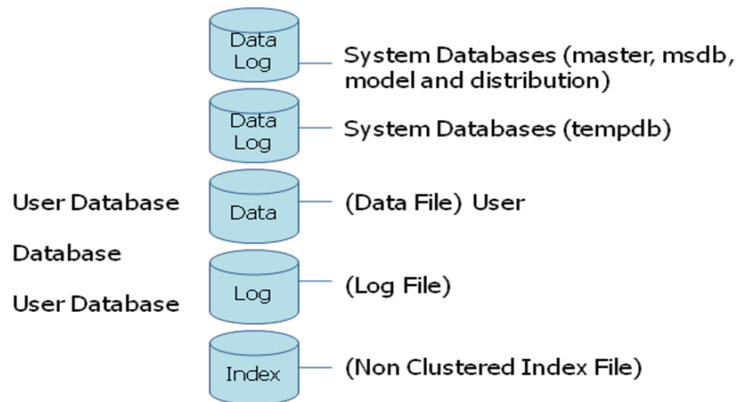


Figure 1-1 Typical Data base File Distribution

Benefits: Assigning user database files to their own dedicated volume (Windows disk) maximizes the efficiency of I/O requests. It also provides performance and space management flexibility.

Assigning the tempdb database files to their own dedicated volume (Windows disk) provides performance and space management flexibility.

Allocating Database File Space

BEST PRACTICE: Preconfigure user database file size allocation and automatic file growth values. Also preconfigure tempdb database automatic file growth values.

When: Allocate file space size when you create the user database. You also can designate file space at planned intervals as needed. For the specific tempdb database, wait until after installation.

How: SQL Server 2005 includes the option to set a predefined value for the automatic growth of database files. Set the value for automatic file growth to give the necessary growth in as few increments as possible. Minimizing the number of increments minimizes the use of disk resources.

For example, if a database needs to grow in size by 1Gb due to the workload and the automatic file growth value is 100MB, in order to achieve a growth of 1Gb, the database has to auto-grow in 100 MB increments for ten times. These excessive operations are too resource-intensive because there can be many other operations that are required of the disk at the same time, such as data inserts, deletes, and so forth. An automatic file growth value of 500MB requires only two incremental auto growths, so it will be less resource-intensive.

Therefore, consider such factors as database workload and disk space—present and future—when setting the value for automatic file growth.

Benefits: Defining specific user database file sizes helps achieve uniform disk space

use.

Defining automatic file growth values for the user database size helps minimize disk resource use.

Defining automatic file growth values for the tempdb database size helps minimize disk resource use.

Configuring Automatic Alerts with the SQL Server Agent

You can use the SQL Server Agent to automate many database administration jobs by allowing you to schedule your own jobs and scripts. One important job SQL Server Agent can do is to automatically notify relevant personnel when certain thresholds have been reached and specific events occur. Otherwise, you will not know the risks to

the system until users begin to report problems. These problems can range from security concerns to performance issues. Configuring automatic alerts requires configuring the three components of the SQL Server Agent: the operators, the alerts, and the history. These components can be configured in any order.

Configuring the Operators

Operators identify that person or those persons to whom alerts should be sent. Operators also designate whether alerts should be sent by email or to a cell phone or pager.

BEST PRACTICE: Configure the system operators that will receive alerts.

When: Configure the operators immediately after installation.

How: Configure the operators in the SQL Server 2005 Enterprise Manager GUI in the operators object.

Benefits: Configuring the operators to receive alerts is a proactive measure to aid and automate the monitoring of alerts that occur in the SQL Server 2005 instance.

Configuring the Alerts

Alerts can be triggered by performance or by events. Triggers are based upon thresholds that you set.

BEST PRACTICE: Set up alerts to monitor the database while it is unattended. System alerts that affect resources such as memory, processor utilization, file growth, and so forth are especially important.

When: Configure alerts immediately after installation.

How: Configure alerts in the SQL Server 2005 Enterprise Manager GUI in the SQL Server Alert object.

Benefits: Setting up alerts is a proactive measure to aid and automate the monitoring of database events and behavior.

Configuring the History

Most of the properties of the jobs can be configured as you see fit. However, collecting too much job history can cause problems.

BEST PRACTICE: Restrict the amount of job history that is collected by the SQL Server Agent.

When: Configure the jobs immediately after installation.

How: Configure the job history in the SQL Server 2005 Enterprise Manager GUI in the SQL Server Agent object properties.

Benefits: Setting a specific size limit on the job history helps to make sure that the msdb system database does not become unmanageable in size.

Configuring Auto Shrink

Although it is wise to conserve database space, Auto Shrink is not the preferred method.

BEST PRACTICE: Set the Auto Shrink option to false.

When: Set the Auto Shrink option immediately after installation.

How: Set the Auto Shrink option in the SQL Server 2005 Enterprise Manager GUI in the SQL Server database property.

Benefits: Setting the Auto Shrink option to false for a database helps minimize resource conflicts between the database and the SQL Server 2005 instance. Using manual controls instead gives you more control over database size management and when database shrinking should occur.

For a detailed explanation of why you should not use Auto Shrink, see "Turn Auto Shrink OFF!" by Paul Randal, November 12, 2007. Go to:

<http://www.sqlskills.com/blogs/paul/CategoryView,category,Shrink.aspx>

Backing Up and Restoring Files

BEST PRACTICE: Store the database backup files separately from the online database files. Store the backup files on a separate, dedicated storage location for testing backup file restores and disaster recovery.

When: After installing and configuring SQL Server 2005, set up the backup function and the restore function.

How: You can configure the backup file target with the SQL Server 2005 Enterprise Manager GUI for backup devices, maintenance plans, and T-SQL backup command use.

Benefits: Storing the database backup files separately from the online database files minimizes the risk of a single point of failure in the event of a system outage.

BEST PRACTICE: Test the restore and consistency of backup files.

When: Test the backup and restore process on a regular basis. Testing frequency

depends on organizational needs for data availability and data recovery.

How: To test and validate the integrity of backup files, conduct a database restoration in a test environment. After a restoration, execute the SQL Server 2005 Database Consistency Check (DBCC) command against the restored database to detect any consistency errors.

SQL Server 2005 offers different recovery model options for databases. Before choosing a model, assess the system environment and the business needs of the organization regarding data use, data availability, and data recovery. For more information about SQL Server 005 database recovery models, go to:

<http://www.microsoft.com>

Benefits: Performing a database restore and consistency check in a test environment validates backup files and database consistency.

Restoring the Database Owner

By default, the database owner is the database creator. Therefore, when the database is restored to a different server, the database owner resets. The new database owner could be different than you intended.

BEST PRACTICE: Always restore the database owner after database restore to a different server. To restore the database owner, make sure that the original database owner still has a valid login on the target server where the database was restored.

When: Always validate the database owner login after restoring a database to a different server.

How: When a database is restored to a different server, the database owner defaults to the same login that was valid during the restoration. To validate the database owner login, go to the SQL Server 2005 Enterprise Manager GUI in the SQL Server database property. To change the owner of a database, use the SQL Server 2005 system stored procedure, `sp_changedbowner`. For more information, refer to Microsoft SQL Server documentation.

Benefits: Restoring the database owner makes sure that the original database owner can still log in to the restored database in its new location on the target server.

Scheduling Tasks That Result in Whole Procedure Cache Flushing

The SQL Server 2005 system automatically flushes the whole procedure cache after certain database-level operations. To avoid the problems associated with whole procedure cache flushing, carefully schedule these tasks.

BEST PRACTICE: Perform certain database-level operations during non-peak production hours.

When: When the SQL Server 2005 instance is serving a production environment, you must perform the following tasks during maintenance windows and off hours.

- Detach a database.
- Drop a database.

- Change a database using specific options of the ALTER DATABASE command.
- Restore a database.

How: For more information, including a list of Alter Database command options that are affected, consult the Microsoft support web site. Go to:

<http://support.microsoft.com/>

Benefits: When a database is detached, dropped, restored, or the alter database command is used with specific options, the SQL Server 2005 whole procedure cache is cleared. Clearing the cache can cause a sudden decrease in performance with queries and stored procedures.

Performing these tasks during non-peak production hours helps to prevent queries and stored procedures from being affected by the whole procedure cache being flushed at once. The flushing causes cache recompiles that can contribute to poor query performance.

Using Standard Naming Conventions

BEST PRACTICE: Adhere to naming standards for databases, objects, and instances.

When: Always use naming standards when managing the various components of a SQL Server 2005 instance, such as databases and objects (tables, stored procedures, functions, views and so forth). Consistently use and enforce naming standards 100 percent of the time.

How: Table 1-2 shows the various components and gives examples for naming standards.

Table 1-2 Suggested Naming Conventions

Component	Prefix	Name	Notes
Job	job	job + <description>	Making the prefix lower case enhances readability by distinguishing the prefix from the remainder of the filename.
Alert	alert	alert + <description>	
Report	rpt	rpt + <description>	
Table	t	t + <description>	
Trigger	tr	tr + <table name> + < description>	
Function	f	f + <description>	
Stored Procedure	p	p + <primary table name> + <description>	
View	v	v + < primary table name> + <description>	

Index <unique constraint>	ak	ak + <table name> + <column name>	
Index	ix	ix + <table name> = <column name>	
Primary Key	pk	pk + <table name>	
Foreign Key	fk	fk + <table name> + table name>	Parent/child table
Table Columns		<purpose description>	
Backup File	<database name>	<database name> + <file group name> + <date/time stamp>	Complete backup
Backup File <File Group>	<database name>	<database name> + <file name> + <date/time stamp>	File group backup
Backup File <File>	<database name>	<database name> + <file name> + <date/time stamp>	File backup
Database		<purpose description>	
Instance		<name based on business rules>	Upper case

Benefits: Establishing naming standards helps to differentiate your custom objects from those objects that are products shipped by SQL Server 2005. Naming standards also aide in troubleshooting and manageability across different databases within the same instance.

Conclusion

SQL Server 2005 provides many features to meet storage, performance, and availability needs relating to data management. It is essential that the initial implementation be planned out and tested for the environment. This document has addressed items to consider when implementing and configuring SQL Server 2005 to build a solid foundation. Moving forward, creating a framework that involves testing and change management will bring continuous improvement to managing data in the organization.

Glossary

Auto Shrink

An algorithm that reduces the size of database files by compressing them.

Authentication Mode

The method by which a person's identity is verified for security purposes.

Collation

The assembly of written information into a standard order. The most common collation methods are alphabetical and numerical, but special sets of information can require unique sorting methods. Collation allows the seeker to find information in a predictable location.

Cluster on Disk

A cluster unit for disks of the Windows operating system. Made of multiple sectors, the cluster on disk must be formatted with a specific file system—in this case, NTFS. You specify the cluster size, also known as Allocation Unit. The file system uses this multiple of the disk sector size to manage disk space in a Windows operating system. For more information, go to:

<http://support.microsoft.com/kb/140365/EN-US/>

Database Owner

An individual account with full security privileges to the database. Multiple database owners can exist. The *database owner* is separate and distinct from the *database owner role* (db_owner).

DiskPart

A command line tool utility that is included in Windows Server product offerings and is also available to download from Microsoft.

Extent

SQL Server 2005 unit of data storage that consists of eight contiguous pages and is 64KB in size.

File growth increment

The amount of space added to the file every time new space is required.

I/O request

(Input/Output) reading or writing data between disk and memory.

Kerberos

Security authentication process to verify identity.

OLTP

Online Transaction Processing that consists of random read/write workloads by application and ad-hoc usage of a relational database management system.

OLAP

Online Analytical Processing that consists of a read workload where the write activity is scheduled at planned intervals by an extract, transform, and load process or within batches of a relational database management system.

Page

SQL Server 2005 unit of data storage that is 8KB in size.

Sector

Block size of disk storage in a Windows operating system.

Segment

The size of data that a controller writes on a single physical drive before writing data on a different physical drive.

Whole Procedure Cache Flush

Clears the cache of all stored procedures without distinction as to how often or how recently a procedure is used. Any procedure that should continue must be rebuilt after a whole procedure cache. This leads to a decrease in query performance while the cache builds up again.

Document Description

This document gives best practices that can be implemented When: installing SQL Server 2005. Implementing these best practices during installation will build a foundation for storage, performance, and assessment when using SQL Server 2005.

Ownership of Materials

The Document is provided as a courtesy to customers and potential customers of LSI Corporation ("LSI"). LSI assumes no obligation to correct any errors contained herein or to advise any user of liability for the accuracy or correctness of information provided herein to a user. LSI makes no commitment to update the Document. LSI reserves the right to change these legal terms and conditions from time to time at its sole discretion. In the case of any violation of these rules and regulations, LSI reserves the right to seek all remedies available by law and in equity for such violations. Except as expressly provided herein, LSI and its suppliers do not grant any express or implied right to you under any patents, copyrights, trademarks, or trade secret information. Other rights may be granted to you by LSI in writing or incorporated elsewhere in the Document.

Trademark Acknowledgments

SANtricity is a registered trademarks of LSI Corporation. All other brand and product names may be trademarks of their respective companies.

Performance Information

Performance tests and ratings are measured using specific computer systems and/or components and reflect the approximate performance of LSI products as measured by those tests. Any difference in system hardware or software design or configuration may affect actual performance. Buyers should consult other sources of information to evaluate the performance of systems or components they want to purchase.

Disclaimer

LSI has provided this Document to enable a user to gain an understanding of SQL Server 2005 ("the Tool") when used in conjunction with LSI Storage Systems. This Document and the Tool referenced in it are designed to assist a user in making a general decision as to whether an LSI Storage System configuration is appropriate for such user's objectives. Neither this Document nor the Tool are designed or intended to guarantee that the configuration a user chooses will work in a specific manner. While the guidance provided by this Document and the Tool can help a user to choose an appropriate configuration (or avoid a configuration that is not appropriate), there is no way LSI can guarantee the exact performance and/or results of the Information contained in this Document. Accordingly, LSI assumes no obligation whatsoever for the use of the Information provided in this Document or the Tool, AND UNDER NO CIRCUMSTANCES WILL LSI OR ITS AFFILIATES BE LIABLE UNDER ANY CONTRACT, STRICT LIABILITY, NEGLIGENCE OR OTHER LEGAL OR EQUITABLE THEORY, FOR ANY SPECIAL, INDIRECT, INCIDENTAL, PUNITIVE OR CONSEQUENTIAL DAMAGES OR LOST PROFITS IN CONNECTION WITH THIS DOCUMENT OR THE TOOL.

THE INFORMATION AND MATERIALS PROVIDED IN THIS DOCUMENT AND THE TOOL ARE PROVIDED "AS IS" AND LSI MAKES NO WARRANTIES EXPRESS, IMPLIED OR STATUTORY, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE WITH RESPECT TO THE SAME. LSI EXPRESSLY DISCLAIMS ANY WARRANTY WITH RESPECT TO ANY TITLE OR NON-INFRINGEMENT OF ANY THIRD

PARTY INTELLECTUAL PROPERTY RIGHTS, OR AS TO THE ABSENCE OF COMPETING CLAIMS, OR AS TO INTERFERENCE WITH USER'S QUIET ENJOYMENT.