Best Practices for PTC Windchill on Microsoft SQL Server

Production-proven content and process management software

Author: Ken Lassesen
Reviewers: Tim Atwood (PTC), Victor Gerdes (PTC), Richard Waymire (Solid Quality Mentors)

Abstract

PTC Windchill is the only Product Lifecycle Management (PLM) solution designed from the ground up to work in an Internet-based, distributed-design environment. Whether you need core product data-management capabilities, optimization of processes to meet industry-specific requirements, or support for global product development, Windchill has the capabilities you need for effective management of global product development teams. Microsoft® SQL Server® provides an ideal database platform for Windchill. With SQL Server as a foundation, Windchill can further reduce the time and costs related to managing product development.

This white paper provides best practices for configuring and running Windchill on the SQL Server database platform. The information in this paper complements the detailed support documentation provided on the PTC support Web site. Implementing these best practices can help you avoid or minimize common problems and optimize the performance of Windchill on SQL Server so that you can effectively manage your resources, reduce operating expenses, increase productivity, and improve employee satisfaction.

Copyright Information

Table of Contents

FIGURES

TABLES

Overview

For manufacturers, there is constant pressure today to compete in rapidly changing markets. Companies must be innovative in keeping costs low while maintaining high-quality results. The stakeholders involved in the product development process have also changed. Teams today extend far beyond the central engineering department to include globally dispersed cross-functional groups working on hundreds of products with thousands of parts. The result is an enormous amount of data.

To stay ahead of the global competition, companies find that they must create a collaborative environment that brings together engineering, manufacturing, marketing, and sales teams. Product Lifecycle Management (PLM) solutions can help manufacturers achieve this collaboration, while also streamlining operations and keeping costs down.

PLM is the process of managing all phases of product development—from initial concept through end of life. Effective PLM combines information, methodology, and available resources for each phase of a product's lifecycle, improving a manufacturer's ability to respond swiftly and effectively to changes, new markets, and competitors.

Introducing PTC Windchill: A Complete Solution for PLM

PTC Windchill provides a complete family of solutions for content and process management, helping manufacturers efficiently control all information assets related to product development while optimizing associated business processes. Windchill is the only PLM solution designed from the ground up to work in an Internet-based, distributed-design environment.

Windchill technology forms a solid foundation for a variety of packages that PTC offers to address data, change, configuration, and process management; product development collaboration; project management and execution; and the release of product information to manufacturing management systems. For example,

- **Windchill PDMLink** consolidates scattered islands of product content into a single information source, which can help bring order to chaotic product development processes such as change management and speed the development of new product configurations.
- **Windchill ProjectLink** creates a virtual workspace that becomes the central access point for a project, enabling team members to collaborate with access to the same information. By automating project management activities, Windchill ProjectLink helps customers better manage all of their programs, project schedules, information, and processes.

As an integral component of PTC's Product Development System (PDS), Windchill manages all product content and business processes throughout a product's lifecycle. Windchill connects seamlessly to Pro/ENGINEER for three-dimensional (3-D) computer-aided design (CAD) models, ProductView® for advanced mock-up and interactive visualization, Mathcad® for engineering calculations, and Arbortext® for dynamic publishing.

SQL Server: An Enterprise-Ready Database Platform for Windchill

Microsoft SQL Server provides an ideal database platform for Windchill. SQL Server is a high-performance, integrated database and business intelligence (BI) solution for data management and analysis. This easy-to-implement, easy-to-support foundation provides a multifunctional solution for large-scale online transaction processing (OLTP), data warehousing, and e-commerce applications and a solution for data integration, analysis, and reporting.

SQL Server can help companies manage large volumes of mission-critical data and run software applications—such as PTC Windchill—to optimize their business performance. SQL Server can extract and transform data from a variety of sources, including XML data files, flat files, and relational data sources, and then load it into one or more destinations. In addition to rapid data mining, analysis, processing, and reporting capabilities, SQL Server has built-in features that give you a secure, reliable, and productive data management environment that truly protects your data. With its scalable infrastructure, SQL Server has the capability to grow with your business and keep up with your toughest data challenges.

PTC and Microsoft: A History of Innovation

PTC and Microsoft deliver complementary product development solutions that organizations can use broadly across their infrastructure. With a Microsoft IT infrastructure, you get an open, extensible platform and a simplified user experience. You can take advantage of your existing IT investments for a lower total cost of ownership (TCO). PTC provides full integration with your Microsoft infrastructure, including Microsoft® Office SharePoint® Server, Microsoft® Office Project Server, Microsoft® Office Communications Server, Windows Server®, and SQL Server. The PDS architecture gives you end-to-end solutions for product development and a single source for product and process knowledge.

Together, PTC and Microsoft deliver powerful product development solutions that provide tremendous customer value.

Better Together: Windchill on the SQL Server Database Platform

Running Windchill on SQL Server delivers measureable value by channeling data into manageable, automated processes. This decreases administrative time, improves productivity, reduces costs, and generates greater employee satisfaction. Benchmarking tests confirm that SQL Server scales to meet the performance needs of even the largest enterprise customers, while providing lower initial costs and licensing fees.[1]

Benchmark test results showed that Pro/ENGINEER and Windchill 9.1 performed up to 50 percent faster on SQL Server 2005 than on a competitor's database, with an average performance advantage of approximately 10 percent. These results confirm SQL Server 2005 as a superior database choice for Windchill 9.1.

---

[1] "Microsoft SQL Server: Improved Performance for PTC Windchill," http://www.cio.com/white-paper/588485/Microsoft_SQL_Server_Improved_Performance_for_PTC_Windchill

Best Practices for Installing Windchill on SQL Server

Best practices define the most efficient use of computing resources to produce the best system performance and most reliable delivery and availability of data. Because many factors can contribute to poor system performance, you should aim to implement best practices in both planning and operating your Windchill system. The Windchill application database needs periodic maintenance and tuning. Without such maintenance, database operations can become considerably slower over time.

Implementing best practices helps ensure that your SQL Server database is:
- Configured for administrative monitoring
- Free of database corruption and system errors
- Available when users need it
- Tuned for optimal performance, letting users receive the data they request in a reasonable period
- Able to return to normal operation quickly after a failure

Best practices for Windchill on the SQL Server database platform encompass both hardware and software. They include sizing, setting up, validating, maintaining, monitoring, and backing up your system. The guidance provided in this paper can get you started on items not covered in the following PTC documents: "Windchill on Microsoft SQL Server Installation Planning Guide®: Enterprise Deployment Resource, Windchill® 9.0 & Windchill® 9.1" and "Windchill® and Pro/INTRALINK® 9.0 and 9.1 Server Hardware Sizing Guidelines – Microsoft Windows Platform." PTC and Microsoft provide a great deal of valuable guidance for keeping your Windchill management system running smoothly. For more detailed information, see the Links for Further Information at the end of the paper.

Trouble-free performance is impossible to guarantee. However, if you find that you need to make a service call, ensure that you have first run through the recommended monitoring guidance in the sections that follow. These best practices will provide you with valuable information that you can use to get your system back up and running quickly.

Select, Size, and Configure SQL Server for Optimal Performance

PTC requires SQL Server 2005 Standard or Enterprise Edition for Windchill 9.1. In most cases[2], the recommended configuration is SQL Server 2005 Service Pack 3 (SP3) running on x64 hardware. Larger sites may require Windows 2008 R2 Data Center Edition with SQL Server 2005 Enterprise Edition to support more than eight (8) x64 sockets. The SQL Server 2005 Features Comparison page provides more information, and SQL Server Books Online's Editions and Components of SQL Server 2005 gives you additional guidance on choosing appropriate SQL Server editions and features for your implementation.

---

[2] Running AWE with SQL Server 2005 x86 on an x64 system is not recommended. SQL Server will consume all of the memory except 128MB, which can impact performance. AWE can use only 16GB of memory.

This document walks you through the key steps in setting up SQL Server 2005 for Windchill 9.1.

Note: Windchill 10.0 will run on SQL Server 2008; Windchill 9.1 requires SQL Server 2005.

Hardware Selection

The sizing tables in the Appendix augment the guidance in "Windchill® and Pro/INTRALINK® 9.0 and 9.1 Server Hardware Sizing Guidelines – Microsoft Windows Platform" by showing the suggested initial number of cores, sockets, and operating system (OS) options for running Windchill on SQL Server. Except for the smallest installations (four CPUs or fewer), you will need to run SQL Server Enterprise Edition using x64 CPUs. This paper focuses on this type of configuration. If you are purchasing hardware, you may want to ensure that the hardware will support more cores and more memory then stated in the sizing table. It is considerably more expensive and difficult to replace the hardware later. Many firms deem additional memory and CPUs for existing hardware to be *expenses* while replacement machines may be *capital investments*.

CPUs

Look for CPUs and disk drives with larger caches. Look for CPUs with at least 16MB L3 Cache. For disk drives, require at least a 16MB read buffer. Avoid IA64 CPUs because Microsoft is discontinuing support for them[3]. Do not view a hyper-threading core as two cores. Intel's paper "How to Determine the Effectiveness of Hyper-Threading Technology with an Application" indicates that the actual gain from hyper-threading is only 15%-30% more than a single core.

Memory

The system memory should be the fastest available (DDR3-1600 or better). The motherboard, memory, and CPUs must be compatible.

Physical Hard Drives

Obtain the best performing drives that you can afford. Performance is typically a function of rotations per minute (RPM) and cache size, but interface cards and drivers are involved. The performance of a RAID array is often more dependent on the RAID card than on the hard drives. If not all of the drives are identical, evaluate each hard drive's performance using SQLIO before allocating them to logical drives. (For details, see the comprehensive companion white paper "Best Drive Configuration Practices for PTC Windchill on Microsoft SQL Server," available on the Microsoft and PTC Alliance Web page.)

Before evaluating drive performance, reformat all of the hard drives used for data storage. If you are using Windows Server 2003, make sure you align disk partitions[4]. Windows Server 2008

---

[3] http://blogs.technet.com/windowsserver/archive/2010/04/02/windows-server-2008-r2-to-phase-out-itanium.aspx

[4] http://msdn.microsoft.com/en-us/library/dd758814.aspx

automatically does disk alignment, 2003 does not. Incorrect alignment can cut performance by up to 30%[5].

If you have physical drives with different performance characteristics, you should assign them to different components as suggested in Table 1.

Table 1 - Assigning Components by Disk Performance

Random I/O Drive Performance	Use for	Description
Best	Tempdb	This is where most performance-critical activities occur.
Medium	Random access tables	This type of activity requires frequent movement across the disk.
Worst	Log files, log tables	This type of component serializes writes to the disk into a continuous stream. The serial write performance should *not be less* than the random write performance of the database being logged.

A Storage Area Network (SAN) is often the preferred storage medium because it is the easiest scale-out solution[6]. A SAN Logical Unit Number (LUN) is equivalent to a RAID unit and behaves like a logical drive. SAN units should be on isolated networks supporting at least 1GB transfer speeds.

Here are PTC's recommendations for hard drives:
- Do not use software to emulate RAID; always use hardware.
- Do not use SATA drives; use SCSI drives.
- Use RAID 1+0 when possible; otherwise, use RAID 1, RAID 5, or RAID 6.
- Use more small drives instead of fewer large drives. Two small 7200 RPM drives striped may deliver better performance than one 15000-RPM drive and at substantially less cost.

In addition to the "Best Drive Configuration Practices" paper, see Storage Top 10 Best Practices and Mike Ruthruff's Predeployment I/O Best Practices for more details.

Configuring Tempdb

The SQL Server tempdb system database provides temporary storage space for many things from temporary tables to DBCC CHECKDB worktables. Tempdb configuration greatly influences Windchill performance. SQL Server will use tempdb for various activities if there is insufficient memory on your system, and this shifting of activity to tempdb results in a significant loss of performance. A poorly configured tempdb compounds the performance loss.

---

[5] http://www.brentozar.com/archive/2009/01/partition-alignment-in-virtual-machines/

[6] http://msdn.microsoft.com/en-us/library/aa479364.aspx

For all disk drives, PTC recommends the following:
- The drives should be RAID (1 or 1+0, but not RAID 5).
    - If one of the drives should fail, SQL Server will shut down with RAID 1.
- The drives should be the fastest (typically 15000 RPMs) and provide the highest throughput that you can afford.
    - Drives should have write caching turned off[7].
- Verify that the disks' partitions are aligned properly.
- The Windows swap file should not be located on any of the above drives, nor on the OS drive.
- Do not change the Windchill database owner from system administrator (sa).
- Create alerts using SQL Server Agent to track for errors 1101 or 1105, which indicate that you are running out of space on at least one drive.
    - Error 1101: Could not allocate a new page for database because of insufficient disk space in filegroup.
    - Error 1105: Could not allocate space for object.

For tempdb configuration, PTC recommends the following:
- Tempdb should *not* be located on:
    - The OS drive
    - Any drive that also contains the database files
- Tempdb should be located on at least one dedicated logical drive.
    - Tempdb should be striped across as many drives as practical.
- Locate the tempdb log on a dedicated logical drive. There is rarely benefit to spreading the transaction log across multiple drives.
- Create one tempdb data file per core for the tempdb filegroup.
    - All of the files in the tempdb filegroup should be the same size[8].
    - Use striped drives to increase throughput.
    - If distributing the files across different physical drives, the drive speeds should be the same.
    - See Table B in the Appendix for more details.
- Use Table B in the Appendix to determine the number of tempdb files that should be in each filegroup.
    - Enable autogrowth and set it to 10%[9] to protect system performance if your environment does not conform to the capacity tables.
- Once you have populated the database with the expected volume of data, monitor tempdb while rebuilding the indexes of the table that has the most rows.
- Tempdb should always be on drives with available space being at least 20% of the database size. A larger tempdb may be required depending on the ad hoc queries that

---

[7] http://support.microsoft.com/kb/234656

[8] http://www.db-staff.com/index.php/microsoft-sql-server/65-tempdb

[9] http://msdn.microsoft.com/en-us/library/ms175527.aspx

are executed against the database. A bad query might result in significant growth. Adding an additional large index may result in index rebuilds failing due to a lack of space in tempdb. Running out of space in tempdb may stop Windchill and reduce SQL Server to limited functions.

- Caution: Turning on Snapshot Isolation to match Oracle's default transaction behavior requires appropriate sizing adjustments to tempdb. It may significantly increase the demands on tempdb and, with the wrong hardware configuration, decrease performance.
- If you are using a SAN, tempdb should have its own dedicated drives and its own LUN.

Error! Reference source not found. shows a sample 4-core tempdb configuration following these best practices.

Figure 1 - Example of a 4-core Tempdb Configuration

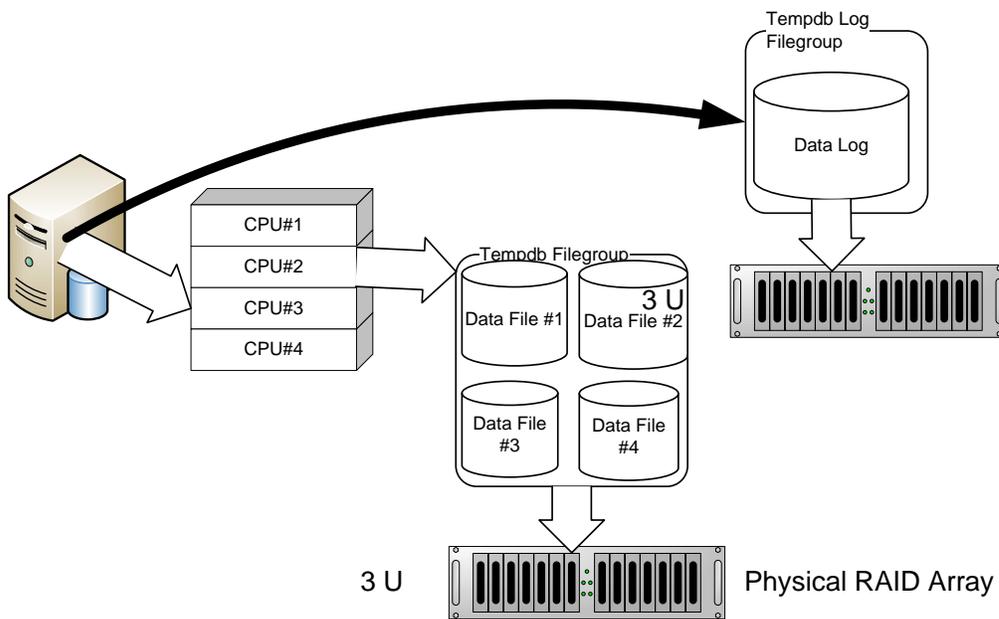Table 2 shows the database options available for tempdb and PTC's recommended values for them to ensure the best performance with Windchill.

Table 2 - Recommended Values for Tempdb Database Options

Database Option	Default Value	Recommended Value
AUTO_CREATE_STATISTICS	ON	ON
AUTO_UPDATE_STATISTICS	ON	ON
AUTO_UPDATE_STATISTICS_ASYNC	OFF	ON
CHANGE_TRACKING	OFF	OFF

Make sure you also follow these recommendations for additional SQL Server options affecting tempdb:
- SORT_IN_TEMPDB should be ON.
- CHECKSUM should be ON.

SQL Server drops tempdb and recreates it every time SQL Server stops and is restarted. It is important to note the size of tempdb before stopping the server and then use ALTER DATABASE to set the initial size of tempdb to this size if it is greater than the current setting. Doing this reduces the number of times that autogrowth occurs, improving performance and reducing disk fragmentation. PTC also recommends instant file initialization of tempdb, which we will cover in a moment.

In addition, as part of regular weekly maintenance, you should:
- Check the SQL Server error log for the following error numbers: 1101, 1105, 3959, 3958, and 3966. Alternatively, set up a SQL Server Agent alert to email a notification when these errors occur. If you see any of these errors, you should troubleshoot for a sizing issue[10].
- Check the size of tempdb versus its startup setting. Increase the startup setting if it is less than the size observed.
    - Record the maximum size of tempdb you see for periodic review.

Instant File Initialization
The default behavior for tempdb is to recreate itself when SQL Server starts with a small database size and grow as needed. During this growth process, SQL Server initializes the file by filling the space with zeros. Specifying a larger initial file size and doing instant file initialization will improve performance. The initial size should be at least the largest size seen in production. See the SQL Server Premier Field Engineer blog post "How and Why to Enable Instant File Initialization" for more information.

Using Tempdb in Memory
SQL Server versions prior to 2005 allowed tempdb to be explicitly placed in memory for better performance. SQL Server 2005 requires the creation of a RAM disk to place tempdb in memory (see Microsoft SQL Server I/O subsystem requirements for the tempdb database). The other option to obtain equivalent performance is to use a DRAM-based solid-state drive.

Putting tempdb on a RAM drive should never be an initial configuration. When your application is fully populated and in regular production, verify the condition below to determine if there is adequate RAM:

```
Total Physical Memory – (1.2 * Maximum tempdb Size Observed > (1.2 * Maximum
Memory Used)
```

---

[10] http://msdn.microsoft.com/en-us/library/ms176029.aspx

This formula includes a 20% buffer to allow for future growth. If you do not have sufficient physical memory to fulfill the above condition, you may have lower throughput if you place tempdb in memory.

For more information about drive configurations, see the "Best Drive Configuration Practices for PTC Windchill on Microsoft SQL Server" and "TEMPDB Capacity Planning and Concurrency Considerations for Index Create and Rebuild."

Memory and CPU

The sizing guidance provided in "Windchill and Pro/INTRALINK 9.0 and 9.1 Server Hardware Sizing Guidelines – Microsoft Windows Platform" may not apply to your installation because of a difference in usage pattern. You can confirm whether your configuration for memory and CPU is adequate by using Windows Performance Monitor.

To get the information you need, configure and run Performance Monitor as follows:
- Set the polling frequency to 30 seconds or more.
- Do not run it on the machine hosting SQL Server.
- Log the data to a physical file.
- Consider running it as a service or at a scheduled time for consistent data collection.
- The monitoring period should be the weekly peak load; typically, this occurs on Thursday afternoons.
- Retain a 24-hour baseline of the busiest day to use for future reference.

After every major software patch or hardware change, rerun Performance Monitor and compare the results to determine the positive or negative impact that the change made. During peak load periods run Performance Monitor and review the items shown in

.

Table 3 - Performance Monitor Indicators of Inadequate Resources

Object	Counter	Description	Recommended Action
Logical Disk	% Free Space		< 15% : More disk space needed
	Avg. Disk sec/Read	Applies to *each* logical drive	> 15: I/O/disk bottleneck[11]
Memory	Cache Bytes		> 300MB: Disk bottleneck, better controllers, faster disks[12]
	% Committed Bytes in Use		> 80%: Add more memory
	Available Mbytes		< 5% of total RAM: Add more memory
Network Interface	Output Queue Length		> 2: Faster network card or segment the network
SQL Server: Buffer Manager	Buffer-Cache Hit Ratio	Ratio of reads found cached in memory	< 95%: Add more memory
	Page Life Expectance	Ratio of reads found cached in memory	< 300 or sudden drops: Add more memory [13]
SQL Server: General Statistics	User Connections		If 255 (system default) or current default, you need to increase setting[14]
SQL Server: SQL Statistics	Batch Request/Sec		If 0.9 >(Batch Request – SQL Compilation) / Batch Requests: CPU bottleneck[15]
	SQL Compilation/Sec		< 90%: CPU bottleneck[16]

---

[11] http://technet.microsoft.com/en-us/library/cc966401.aspx

[12] http://technet.microsoft.com/en-us/magazine/2008.08.pulse.aspx?pr=blog

[13] http://technet.microsoft.com/en-us/library/cc966401.aspx

[14] http://projectserverblogs.com/?p=1378

[15] http://technet.microsoft.com/en-us/library/cc966401.aspx

[16] http://technet.microsoft.com/en-us/library/cc966401.aspx

Additional Performance Analysis with DMVs

In addition to Performance Monitor, you can use SQL Server's dynamic management views (DMVs) to analyze your system's performance. SQL Server tracks wait information into DMVs, and you can execute the following queries against them to detect additional problems[17].

- To detect a CPU bottleneck:
  - ```
    select wait_type from sys.dm_os_wait_stats where
    wait_time_ms > 0 and signal_wait_time_ms/wait_time_ms > .25
    ```
- To see if CPU throughput is being lost to parallelism:
  - ```
    select wait_type from sys.dm_os_wait_stats where
    wait_type='cxpacket' And wait_time_ms > 0 and
    signal_wait_time_ms/wait_time_ms > .10
    ```
- To investigate a possible blocking bottleneck:
  - ```
    select *,wait_time_ms/(waiting_tasks_count+1) as Avg from
    sys.dm_os_wait_stats where wait_type like 'LCK_%' and
    waiting_tasks_count > 0 and max_wait_time_ms > 2 *
    wait_time_ms/(waiting_tasks_count+1)
    ```
- To detect an I/O bottleneck:
  - ```
    select * from (select Top 20 * from sys.dm_os_wait_stats
    order by max_wait_time_ms Desc) TopItems Where wait_type in
    ('ASYNCH_IO_COMPLETION', 'IO_COMPLETION', 'LOGMGR',
    'WRITELOG') or wait_type like 'PAGEIOLATCH_%'
    ```
  - (Also see **Error! Reference source not found.** for an example of a possible problem with the log file; a 0.39 second delay was seen for writing to the log, which could slow the entire database.)

Execute these queries at the end of the business day so that you let the data accumulate beginning from when SQL Server was started. When you want to observe the statistics for a specific period, you can reset the counters using the following command:

```
DBCC SQLPERF ('sys.dm_os_wait_stats', CLEAR);
```

Figure 2 - Example of Possible I/O Bottleneck on the Log File

	wait_type	waiting_tasks_count	wait_time_ms	max_wait_time_ms	signal_wait_time_ms
1	WRITELOG	2853	6281	390	125
2	PAGEIOLATCH_SH	638	1375	140	15
3	IO_COMPLETION	874	1171	140	15
4	PAGEIOLATCH_EX	361	359	46	0
5	PAGEIOLATCH_UP	32	15	15	0

---

[17] http://technet.microsoft.com/en-us/library/cc966401.aspx

Locked Pages

Starting with SQL Server 2005 SP3, Standard Edition supports Locked Pages in memory, which gives similar advantages to running SQL Server 2005 x86 with AWE. This option may improve performance if you have adequate memory. For instructions on enabling this option, see the article "Support for Locked Pages on SQL Server 2005 Standard Edition 64-bit systems and on SQL Server 2008 Standard Edition 64-bit systems[18]."

Lock Escalation

Lock escalation rarely occurs with Windchill, but it may occur with some usage patterns. If you observe blocking conditions, start a SQL Server Profiler trace that includes the Lock:Escalation event. If you see this event, read this support article about how to resolve the problem.

Database Files

Running PTC on SQL Server provides a firm foundation for your mission-critical PLM activities. PTC Windchill supports essential product engineers who cannot do their work if SQL Server is not available. Thus, it is critical that contingency planning be at the forefront of your installation planning. At a minimum, it is strongly recommended that hard drives containing database files be redundant (mirrored or parity) on hot-swappable drives. If one drive fails, there is no loss of data, nor is there a need to bring SQL Server down to replace the drive.

Hardware failures happen, and there are two common paths for recovery:
- **Via an RAID array with hot pluggable drives:** The database will not be offline.
  1. Replace the failed drives promptly.
  2. Rebuild the failed drive dynamically from the other drives.
- **Via the Full recovery model:** If drives are not mirrored or parity-swappable, you must use the SQL Server Full recovery model. The database will be offline until recovered, which may be hours or days depending on the size of the database and the number of backups you need to restore.
  1. Replace the failed drive.
  2. Restore the last full database backup.
  3. Restore the subsequent differential database backups in sequence.
  4. Restore the transaction log backups in sequence since the last differential database backup.
  5. Restore the current transaction log.

---

[18] http://support.microsoft.com/default.aspx/kb/970070

RAID Drive Arrays

There are two important concepts involved with RAID drives:

- Performance—done via striping
- Redundancy—done by mirroring or parity

Table 4 contains a summary of common RAID types and their features. (See "Best Drive Configuration Practices for PTC Windchill on Microsoft SQL Server" for further information.)

Table 4 - RAID Types

Raid Type	Features	Drive Ratio	Description
0	Striped	1:1	Faster speed, no backup
1	Mirror	1:2+	Redundant drives, data not lost if a drive fails
3 and 4	Striped Parity	1:3+	Poor performance, redundant drives, better storage efficiency than RAID 1
5	Striped Parity	1:3+	Can tolerate the loss of 1 drive
6	Striped Dual Parity	1:4+	Can tolerate the loss of 2 drives
10 or 1+0	Mirror Striped	1:4+	First mirrored, then striped
01 or 0+1	Striped Mirror	1:4	Same as above except the sequence is reversed

Note: A logical drive (M:) that is RAID 1+0 will consist of at least four (4) physical drives. For most of this document, a drive means a *logical* drive. Be alert for multiple logical drives partitioned from one physical drive—it may result in loss of both performance and redundancy.

SAN Drives

If you are using SAN drives for your database files, some special settings affect performance. In particular, the Host Bus Adapter default is too low a value; use the recommendation shown in Table 5. SQLIO provides guidance on the optimal settings for your hardware. For more information, see Mike Ruthruff's SQL Server Predeployment I/O Best Practices article.

Table 5 - SAN Setting

Database Option	Default Value	Recommended Value
Host Bus Adapter (HBA) Queue Depth	8-32	64 (Varies by hardware[19])

---

[19] http://blogs.msdn.com/joesack/archive/2009/01/28/sql-server-and-hba-queue-depth-mashup.aspx

Recovery Model

PTC recommends using the SQL Server Full recovery model for Windchill because the cost of losing data is significant, both in terms of recreating the data and in terms of identifying what was lost. This model requires regular transaction log backups, as follows:

- **Transaction logs and backups should never be on a hard disk that contains any Windchill database file—if this drive goes bad, it is impossible to recover**.
- If the transaction logs drive fails (and is not mirrored), then you must immediately:
    o Stop SQL Server
    o Do a full backup
    o Replace the drive
    o Start SQL Server
- You should schedule automatic backups as shown in **Error! Reference source not found.**.

Table 6 - Frequency of Backups by Installation Size

Action	Small Installation	Medium Installation	Large Installation
Full Backup	Weekly	Daily	Daily
Differential Backup	Daily	Every 4 working hours	Every hour
Transaction Log Backup	Every 4 working hours	Every hour	Every 15 minutes

Other recovery models are not appropriate for a variety reasons; for example, neither the Bulk-logged recovery model nor the Simple recovery model fully logs binary large object (BLOB) operations.

Cost of Recovering Instead of Using Hot-Swapped Mirror Drives

The Windchill database can be very large, and a full recovery from backups and transaction logs may take many hours or days. The economic cost of lost productivity can become significant. You should view database recovery as a secondary insurance policy, with the primary insurance provided by redundant hot-swappable drives.

Disk Drive Subsystems

Table 7 shows the expected performance from a variety of disk drive options. The better performance typically comes with higher hardware costs and higher skill levels to correctly configure and maintain the system.

Table 7 - Ranked Performance of Different Disk Drive Subsystems

File Type	DRAM SSD	SAN	RAID 0	RAID 1+0	Raid 5
Tempdb	1	2	3	4[20]	No
Tempdb Log	1	2	3	4	No
Transaction Log	No	1	No	2	3
Database	No	1	No	2	3

Here is a reminder of PTC's recommendations for drives containing data files:
- The logical drives should be RAID 1, RAID 1+0, or RAID 5.
    - Use SCSI hard drives and not SATA physical drives.
    - Use smaller physical drives instead of bigger drives.
    - RAID should be hardware based and not software based.
- The physical drives should be the fastest drives and highest throughput that you can afford. Remember to test the throughput *in situ.*
- Data files should not be located on the drive where the OS resides.
- The Windows swap file should not be located on any of these drives, nor on the OS drive.
- The transaction log files should be on a different logical drive or LUN than any database files.
- When using multiple disks in a filegroup, make sure that all of the files are the same size and the physical drive speeds are the same.
- Verify that the disk partitions are aligned properly.

Windchill Logical Files

The default installation of Windchill creates five (5) logical files, as Table 8 shows. PTC's SQL Server Configuration Utility (SCU) determines the initial size of the database files. SCU allows you to create either a medium-size or large database.

Each logical database file is assigned to a separate filegroup, as follows:
- WcAdmin → PRIMARY filegroup
- WcAdmin_blobs → BLOBS filegroup
- WcAdmin_indx → INDX filegroup
- WcAdmin_wcaudit → WCAUDIT filegroup

In addition, you have the Windchill transaction log (WcAdmin_log) file.

For best performance, at least eight (8) logical drives should be available. These logical drives should be independent—that is, not co-located on the same physical drive.

---

[20] Mirroring is required for high availability systems so that a single physical drive failure does not stop SQL Server.

Table 8 - Windchill Logical Files

Logical Name	Initial Size	Activity Type	Volume	Primary Activity
WcAdmin	1GB	Random	High	R/W
WcAdmin_blobs	1GB	Random	Low	R/low W
WcAdmin_indx	1GB	Random	Medium	R/W
WcAdmin_wcaudit	240MB	Sequential	Low	W
WcAdmin_log	1GB	Sequential	Low	W

Using a tool such as Microsoft's SQLIO, evaluate each logical drive to determine the actual performance for sequential and random access. The allocation of drives should follow this heuristic pattern:

- Because transactions cannot happen faster than they can be logged, the lowest-latency *highest sequential-write* drive(s) should be assigned to **tempdb log**, with preference being given to the highest throughput metrics.
- The lowest latency remaining drive(s) for *random read and write* should be assigned to **tempdb**, with preference being given to the highest throughput metrics.
- The lowest latency remaining *sequential write* drive(s) should be assigned to the transaction log **(WcAdmin_log)**, with preference being given to the highest throughput metrics.
- The lowest latency remaining drive for *sequential write* should be assigned to **Wc_audit**, with preference being given to the highest throughput metrics; audit trails are similar to logs in behavior.
- The lowest latency remaining drive(s) for *random read and write* should be assigned to **WcAdmin_indx,** with preference being given to the highest throughput metrics.
- The lowest latency remaining drive(s) for *random read and write* should be assigned to **WcAdmin**, with preference being given to the highest throughput metrics.
- The remaining drive(s) should be assigned to **WcAdmin_blobs**, with preference being given to the highest throughput metrics.
- Backups should be assigned to whatever is left over (or to network locations).

After installing Windchill on SQL Server, you should adjust the autogrowth properties to reflect your environment. PTC recommends using the largest of the following:

- 10%, as shown in **Error! Reference source not found.**
- Determine the size of the largest document in MB that you expect to store, and multiply that value by 100. Increase **WcAdmin_blobs** autogrowth to either this value (in MB) or to the percentage of the initial size that equals this value. (Never decrease the autogrowth factor.)
- When Windchill is in operation, record the weekly size of each database file and then set autogrowth to twice the average weekly increase. This strategy results in autogrowth happening only twice a month. This may be done by executing the following query:

```
Select Name,(Size * 8)/1024 as MB from sys.database_files
```

Figure 3 - Windchill Database Files Dialog Box Showing Autogrowth
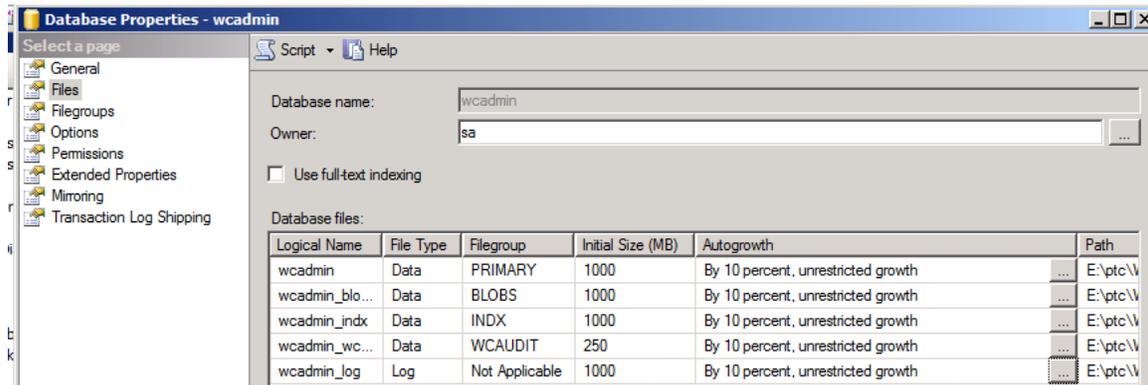
The "Best Drive Configuration Practices for PTC Windchill on Microsoft SQL Server" paper provides detailed analysis for configuring logical drives. Table 9 shows a recommended logical drives configuration.

Table 9 - Example of Logical Drive Assignments

Drive #	Contents
1	Tempdb (database)
2	Tempdb Log (database log)
3	wcAdmin Log (database log)
4	INDX (filegroup)
5	PRIMARY (filegroup)
6	BLOBS (filegroup)
7	WCAUDIT (filegroup)
8	Backups (operating system)

Index Tuning

Tuning and maintenance of indexes can often result in dramatic performance improvements. Indexes fragment from constant inserts, updates, and deletes. Thus, PTC recommends:

- Scheduling weekly rebuilds of all indexes.
    - With SQL Server Enterprise Edition, you can rebuild online.
- Only rebuild immediately before a full database backup and after the last transaction log backup. This will reduce the size of your backups.

What Is a Fill Factor?

A fill factor is the space left in an index record for adding new entries. When a record lacks space, it splits into two, creating two new records that are each 50% filled. This process continues indefinitely.

The locations of the new records are not in sequence (as the original index would be), so the disk drive may have to skip back and forth to find the new split record located somewhere else on the drive and then return to the old location to read the next record. This random (instead of

22

the ideal sequential) reading of the index can slow performance considerably. If you look at a contemporary 15000-RPM SCSI drive specification, you will see 0.3 milliseconds for a single track seek, 4.5 milliseconds for average seek time, and 11 milliseconds for maximum seek time. If 5% of the index records are split records, the time to read the index will double and performance will drop by 50%.

In general, with a new database, a low fill factor (50%) is best because of the high percentage of new records that will occur each day. As the database populates, you should reduce the fill factor because the percentage of new records will reduce dramatically. **Error! Reference source not found.** shows how the percentage of growth decreases over time with the same number of records added weekly.

Figure 4 - Rate of Growth Over Time with Constant Rate of New Records
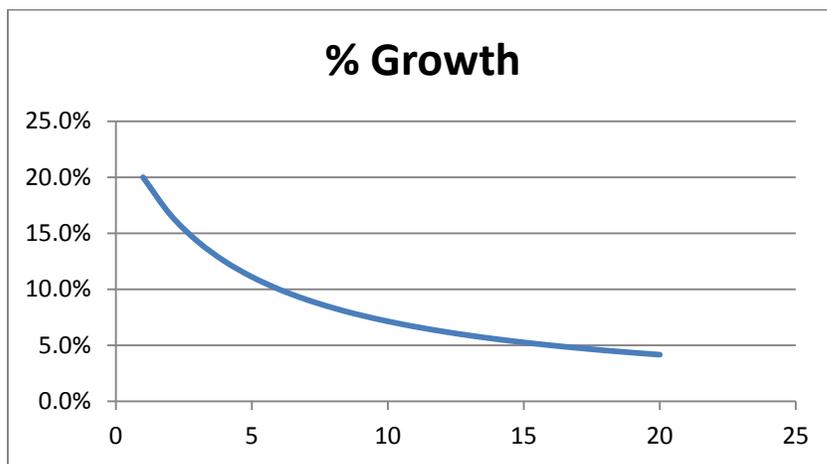
If you keep track of the percentage growth of each table between every regular rebuild, you can use Microsoft Excel to compute fill factors on a table-by-table basis, as Table 10 shows. This computed fill factor should result in less than 2% of index records splitting in a week.

Table 10 - Calculating Fill Factor from Historical Data

Week	Growth	Formula
1	5.7	
2	6.3	
3	4.5	
4	5.9	
5	5.4	
6	4.8	
7	6.8	
Average	5.628571	=Average( )
Std Dev	0.807701	=StdDev( )
Fill Factor	91.94833	=100-Average() – 4 * StdDev( )

Evaluating Index Fragmentation

To obtain information about problem indexes in wcAdmin, you can use this T-SQL code:

```sql
SELECT Object_Name(object_id),
      index_type_desc,
      avg_fragmentation_in_percent,
      avg_fragment_size_in_pages
FROM sys.dm_db_index_physical_stats (DB_ID('wcAdmin'), NULL, NULL,
NULL, NULL)
Where avg_fragmentation_in_percent > 0
And avg_fragment_size_in_pages > 1
And index_type_desc<>'HEAP'
Order by avg_fragmentation_in_percent desc
```

An example of this query's output is shown in Figure 5.

Figure 5 - Example of Retrieving Index Fragmentation Statistics

```sql
SELECT Object_Name(object_id),
    index_type_desc,
    avg_fragmentation_in_percent,
    avg_fragment_size_in_pages
FROM sys.dm_db_index_physical_stats (DB_ID('wcadmin'), NULL, NULL, NULL, NU
Where avg_fragmentation_in_percent > 0
And avg_fragment_size_in_pages > 1
And index_type_desc<>'HEAP'
order by avg_fragmentation_in_percent desc
```

	(No column name)	index_type_desc	avg_fragmentation_in_percent	avg_fragment_size_in_pages
1	HolderToContent	NONCLUSTERED INDEX	80	1.15384615384615
2	ApplicationData	NONCLUSTERED INDEX	75	1.2
3	ApplicationData	NONCLUSTERED INDEX	75	1.2
4	HolderToContent	NONCLUSTERED INDEX	75	1.2
5	StreamData	NONCLUSTERED INDEX	71.4285714285714	1.16666666666667
6	WTPartMaster	CLUSTERED INDEX	66.6666666666667	1.125
7	PhaseSuccession	NONCLUSTERED INDEX	66.6666666666667	1.2
8	ProcessStepTemplateLink	CLUSTERED INDEX	63.6363636363636	1.375
9	PreferenceCategory	CLUSTERED INDEX	62.5	1.33333333333333
10	WfExprRobotTemplate	CLUSTERED INDEX	57.1428571428571	1.16666666666667
11	WTPartUsageLink	CLUSTERED INDEX	55.5555555555556	1.5
12	IterFolderMemberLink	CLUSTERED INDEX	54.5454545454545	1.375
13	WfInternalMethodTempl...	CLUSTERED INDEX	53.3333333333333	1.66666666666667
14	WfExpression	CLUSTERED INDEX	50	1.66666666666667
15	PreferenceInstance	CLUSTERED INDEX	50	1.2
16	ComponentLink	CLUSTERED INDEX	50	1.33333333333333
17	WfAssignedActivityTem...	CLUSTERED INDEX	48.1481481481481	1.8

Performance starts deteriorating at 5%[21], and performance degradation can become very significant at 20%. The numbers reported for the heap index type are meaningless and should be ignored.

With weekly rebuilds and a weekly growth rate of around 2%, using a fill factor from **Error! Reference source not found.** will allow appropriate growth with little fragmentation before the next rebuild. If the period between rebuilds is longer, then the fill factor should be lower to allow more growth without fragmentation. Always do a rebuild in the period *between* the last

---

[21] http://technet.microsoft.com/en-us/library/ms189858.aspx

transaction log backup and a full database backup to avoid backing up the log created from rebuilding the index (which can become very large).

Table 11 - Fill Factors by Index Key Size for 2% Growth

Index Key Size	Fill Factor
474	88%
447	89%
251	91%
161	92%
149	93%
103	94%
88	95%
44	96%

If there is not a significant drop in fragmentation, some possible causes might be:
- The index is less than 1000 pages[22]
- Lack of contiguous pages

Types of Indexes

SQL Server provides several types of indexes.
- **Heap**: A heap occurs on a table *without* a clustered index. Records are located via a triplet {FileId:PageNum:SlotNum}, with records written randomly. You will rarely see a heap index in commercial systems.
- **Clustered index**: With a clustered index, the records are ordered physically on disk by the index, giving the table an explicit order.
  - The next record is often physically located next to the current record, so there is no need to do disk seeks to find the record, resulting in excellent retrieval performance.
  - Only one clustered index may exist per table.
- **Non-clustered index:** With a non-clustered index, the index refers to the clustered index, similar to how an index of a book refers to the page where a term appears.

Missing Indexes

Selecting indexes is an art that must balance the cost of index maintenance versus the performance gain by having an index. PTC has built indexes based on anticipated usage, but actual usage is often different. Frequently seeing full table scans is an indicator of a missing index.

The following query returns the queries with the longest execution time and most frequent execution, which can help you identify where adding an index might be useful:

---

[22] http://msdn.microsoft.com/en-us/library/cc966523.aspx

```sql
SELECT top 100 *
FROM (SELECT
OBJECT_NAME(s2.objectid) as ObjectName,
      DB_Name(s2.dbId) as DatabaseName,
    execution_count,
    total_worker_time,
    total_worker_time/execution_count as Average_WorkerTime,
    (SELECT TOP 1 SUBSTRING(s2.text,statement_start_offset / 2+1 ,
      ( (CASE WHEN statement_end_offset = -1
        THEN (LEN(CONVERT(nvarchar(max),s2.text)) * 2)
        ELSE statement_end_offset END)- statement_start_offset) /
2+1))  AS sql_statement,
    plan_generation_num,
    last_execution_time,
    last_worker_time,
    min_worker_time,
    total_physical_reads,
    last_physical_reads,
    min_physical_reads,
    max_physical_reads,
    total_logical_writes,
    last_logical_writes,
    min_logical_writes,
    max_logical_writes
FROM sys.dm_exec_query_stats AS s1
CROSS APPLY sys.dm_exec_sql_text(sql_handle) AS s2
) A
Where DatabaseName = 'wcAdmin'
And ObjectName is not null
And sql_statement not like '%@%'
And sql_statement not like '%#%'
ORDER BY Average_WorkerTime Desc, execution_count Desc
```

Tools to Help Maintain SQL Server Performance

PTC recommends the installation and use of the following free Microsoft and community products to help maintain the performance of your Windchill system:

- SQL Server 2005 Best Practices Analyzer
- SQL Server 2005 Performance Dashboard
- SQL Server Health and History (SQLH2) Tool
- Performance Analysis of Logs (PAL) tool

For information about using these tools, see the Microsoft and PTC Alliance Web page for "Best Maintenance Tools for PTC Windchill on Microsoft SQL Server 2005," which walks you through building additional indexes to increase performance.

Capacity Planning and Monitoring

Database systems are not "install and forget" systems because databases constantly grow. A new customer may result in rapid growth of Windchill, for example. Thus, it is essential to do capacity planning and monitoring of your Windchill system. Begin by recording the following

items available via Performance Monitor, under SQL Server: Databases object, daily and then weekly into Excel:
- Data File Size
- Percentage Log Used

These numbers will allow you to track growth trends and ensure that adequate disk space is always available.

Common DBCC Commands

SQL Server has a useful set of Database Console Commands (dbcc) that provide access to low-level information about the database for remediation of low-level issues. Dbcc can cause blocking problems so use it when there is the least load on the database. Some of the most useful commands are:

- DBCC CHECKDB
  - Checks database integrity and displays advice on how to fix found problems
  - If running this command affects user performance, trace flag 2528 should be set: (DBCC TRACEON (2528, -1))
- DBCC CHECKCONSTRAINTS
  - Checks table constraints; not done by CHECKDB
- DBCC SQLPERF(LOGSPACE)
  - Lists the states of the logs
- DBCC SHRINKDATABASE('wcAdmin', NOTRUNCATE)
  - Moves the data to the front of the file
    - You rarely need to run this, but if you do run it, rebuild indexes immediately afterward
- DBCC OPENTRAN
  - Checks if there are any open transactions; an open transaction can interfere with the normal operation of the transaction log
- DBCC USEROPTIONS
  - Lists the options in effect; occasionally, gremlins may change the values you have set, resulting in a variety of problems. The expected values are shown in Figure 6.

For more information, see DBCC Commands or DBCC (Transact-SQL).

Figure 6 - Reviewing DBCC USEROPTIONS

	Set Option	Value
1	textsize	2147483647
2	language	us_english
3	dateformat	mdy
4	datefirst	7
5	lock_timeout	-1
6	quoted_identifier	SET
7	arithabort	SET
8	ansi_null_dflt_on	SET
9	ansi_warnings	SET
10	ansi_padding	SET
11	ansi_nulls	SET
12	concat_null_yi...	SET
13	isolation level	read committed snapshot

Common Mistakes

Some SQL Server activities that sound logical do not do what you expect and can cause problems. Here are some general database misconceptions:

- Do not shrink the database – Although it sounds like something you should do, do not shrink the database. This action will often result in index fragmentation and worse performance, as well as SQL Server having to reinitialize the file space when it grows.
- Performing Update Statistics after rebuilding indexes – This operation automatically occurs as part of the rebuilding process.
- Rebuilding indexes immediately after doing a full backup – Rebuilding indexes should occur immediately *before* doing a full backup so that the rebuilding log is never part of a transaction log backup.

Conclusion

The following points sum up the best practices described in this paper:

- Sizing and configuring the hardware:
  - Use adequate cores
  - Ensure adequate memory
  - Use appropriate physical drives and controllers
    - Make sure write caching is turned off
  - Ensure data redundancy of logical drives
  - Use sufficient logical drives to avoid contention issues
  - Perform appropriate assignment and sizing of filegroups to logical drives
- Maintaining the database:
  - Perform at least weekly full backups
  - Reorganize or rebuild indexes weekly
    - Use the optimal fill factor for the index key length
  - Add missing indexes that will improve performance
  - Verify that default settings have not been changed

- Monitoring performance:
  - Use tools to speed regular reviews
  - Track database and log growth

Further details on drive configuration and maintenance tasks and tools are available in the MSDN blog about PTC Windchill on SQL Server as well as in two companion white papers available on the Microsoft and PTC Alliance Web page:
- Best Drive Configuration Practices for PTC Windchill on Microsoft SQL Server
- Best Maintenance Tools for PTC Windchill on Microsoft SQL Server 2005

Links for Further Information

PTC Windchill
- Microsoft SQL Server: Improved Performance for PTC Windchill
- Windchill and Pro/INTRALINK 9.0 and 9.1 Server Hardware Sizing Guidelines - Microsoft Windows Platform

SQL Server information can be found in Books Online:
- SQL Server 2008 Books Online
- SQL Server 2005 Books Online

SQL Server Books Online also includes best practice information in the following articles:
- Best Practices for Replication Administration
- Replication Security Best Practices
- Best Practices for Recovering a Database to a Specific Recovery Point

See the SQL Server Best Practices portal for technical white papers, the SQL Server Best Practices Toolbox, Top 10 Lists, and other resources.

Following is a list of technical white papers that were tested and validated by the SQL Server development team. These can help you learn more about specific SQL Server topics.
- A Quick Look at Serial ATA (SATA) Disk Performance
- Best practices for operational excellence
- Best Practices for Running SQL Server on Computers That Have More Than 64 CPUs
- Best Practices for Semantic Data Modeling for Performance and Scalability
- Checklist: SQL Server Performance
- Checksum Problems, Choosing the Correct Recovery Model and More
- Comparing Tables Organized with Clustered Indexes versus Heaps
- Database Mirroring and Log Shipping Working Together
- Database Mirroring Best Practices and Performance Considerations
- Database Mirroring in SQL Server 2005
- Database Snapshot Performance Considerations Under I/O-Intensive Workloads

- DBCC SHOWCONTIG Improvements and Comparison between SQL Server 2000 and SQL Server 2005
- Description of using disk drive caches with SQL Server that every database administrator should know
- Disk Partition Alignment Best Practices for SQL Server
- FLASH Disk Opportunity for Server-Applications
- How to mirror the system and boot partition (RAID1) in Windows Server 2003
- How to use the SQLIOSim utility to simulate SQL Server activity on a disk subsystem
- How To: Use SQL Profiler
- Identifying and Resolving MDX Query Performance Bottlenecks in SQL Server 2005 Analysis Services
- Implementing Application Failover with Database Mirroring
- Improving SQL Server Performance
- Microsoft SQL Server 2005 Tuning Tips for PeopleSoft 8.x
- Microsoft SQL Server Database Engine Input/Output Requirements
- Microsoft SQL Server I/O Basics (2005)
- Microsoft SQL Server I/O subsystem requirements for the Temp Db database
- Monitor and troubleshoot storage performance
- Partial Database Availability
- Performance of WD 250GB SATA Drives + 3ware Controller
- Physical Database Storage Design
- Precision Considerations for Analysis Services Users
- Pre-Configuration Database Optimizations
- Pre-Deployment I/O Best Practices
- RML Utilities for SQL Server (x64)
- SAP with Microsoft SQL Server 2005: Best Practices for High Availability, Maximum Performance, and Scalability
- SQL Server 2005 Best Practices Analyzer (August 2008)
- SQL Server 2005 Deployment Guidance for Web Hosting Environments
- SQL Server 2005 Performance Dashboard Reports
- SQL Server 2005 Security Best Practices - Operational and Administrative Tasks
- SQL Server 2005 Waits and Queues
- SQL Server 7.0, SQL Server 2000, and SQL Server 2005 logging and data storage algorithms extend data reliability
- SQL Server Best Practices
- SQL Server Best Practices Article
- SQL Server Health and History Tool (SQLH2)
- SQL Server Replication: Providing High Availability Using Database Mirroring

- [SQLIO Disk Subsystem Benchmark Tool](#)
- [Storage Top 10 Best Practices](#)
- [Support WebCast: How to Effectively Use SQL Server Profiler](#)
- [Technical Note #28: Common QA for deploying SQL Server ...](#)
- [TEMP DB Capacity Planning and Concurrency Considerations for Index Create and Rebuild](#)
- [Top Tips for Effective Database Maintenance](#)
- [Troubleshooting Performance Problems in SQL Server 2005](#)
- [Uncover Hidden Data to Optimize Application Performance](#)
- [Understanding Logging and Recovery in SQL Server](#)
- [Understanding SQL Server Backups](#)

Bibliography

- *SQL Server MVP Deep Dives* (Manning, 2010), Paul Nielsen, Kalen Delaney, et al.
- *Professional SQL Server 2005 Performance Tuning* (Wiley Publishing, 2008), Steven Wort, Christian Bolton, et al.
- *Inside Microsoft SQL Server™ 2005: The Storage Engine* (Microsoft Press, 2006), Kalen Delaney
- *Professional SQL Server 2005 Administration* (Wiley, 2006), Brian Knight, Ketan Patel, et al.
- *SQL Server 2005 Bible* (Wiley, 2006), Paul Nielsen

Appendix: Hardware Sizing Guide for Windchill 9.1 and SQL Server 2005

Following are guidelines for determining server requirements for both the application tier and the database tier on a Windows Server platform for a typical Windchill PDMLink, Windchill ProjectLink, or Pro/INTRALINK 9.0 or 9.1 installation that uses SQL Server 2005 as the database platform. These guidelines build upon Table 2 in "Windchill and Pro/INTRALINK 9.0 and 9.1 Server Hardware Sizing Guidelines – Microsoft Windows Platform."

Table A lists the required number of CPU cores (or sockets) and the amount of RAM for a database tier running Microsoft SQL Server 2005, based on the weighted number of active users. (See the PTC guidelines for the definition of Active Users.)

Table A uses the term "core" to refer to a single core. The recommendations do not refer to the number of dual-core or quad-core processors placed in the motherboard sockets. Windows Server 2008 licensing is dependent on the number of *sockets* and not the number of *cores*. A Windows Server 2008 Standard Edition will support only 4 cores if the CPUs are single core or 16 cores if the CPUs are quad core.

Do not count hyper-threading cores as two (2) cores; common SQL Server practice is to disable the hyper-threading cores. Enabling hyper-threading results in a maximum theoretical benefit of around 20% more throughput.

Table A - Database Server Sizing for Windchill 9.0 and 9.1

Weighted Number of Active Users	Required Number of Cores	Required Amount of RAM (GB)	Windows Server 2008 Edition (Cores in Socket)[23]	Windows Server 2003 Edition[24]
10	1	2	Foundation	Standard
25	1	4	Foundation	Standard
50	2	2/4	Foundation (Dual)	Standard
100	2	8	Foundation (Dual)	Enterprise
200	2	8	Foundation (Dual)	Enterprise
300	2	8	Foundation (Dual)	Enterprise
400	3	8	Foundation (Quad)	Enterprise
500	3	8	Foundation (Quad)	Enterprise
600	4	12	Standard	Enterprise
700	4	12	Standard	Enterprise
800	5	12	Standard (Dual)	Enterprise
900	5	12	Standard (Dual)	Enterprise
1,000	6	16	Standard (Dual)	Enterprise
1,200	7	16	Standard (Dual)	Enterprise
1,500	8	16	Standard (Dual)	Enterprise
2,000	12	20	Standard (Quad)	Datacenter
2,500	14	24	Standard (Quad)	Datacenter
3,000	17	32	Enterprise (Dual)	Datacenter
3,500	20	32	Enterprise (Quad)	Datacenter
4,000	23	32	Enterprise (Quad)	Datacenter
4,500	25	32	Enterprise (Quad)	Datacenter
5,000	28	32	Enterprise (Quad)	Datacenter

* If the weighted number of active users falls between two values, round up.

---

[23] http://technet.microsoft.com/en-us/library/dd459191(WS.10).aspx

[24] http://technet.microsoft.com/en-us/library/cc758523(WS.10).aspx

Table B - Number of Physical Files in the Tempdb Filegroup

Weighted Number of Active Users	Number of Tempdb Files
10	1
25	1
50	2
100	2
200	2
300	2
400	3
500	3
600	4
700	4
800	5
900	5
1,000	6
1,200	7
1,500	8
2,000	12
2,500	14
3,000	17
3,500	20
4,000	23
4,500	25
5,000	28