



# **Technical report:** **Microsoft SQL Server 2000/2005 and** **IBM System Storage N series**

*Best practices*

• • • • • • • • •

*Document NS3431-0*

October 3, 2007



## Table of contents

<b>Abstract</b> .....	<b>4</b>
<b>Introduction</b> .....	<b>4</b>
Purpose and scope .....	4
Intended audience .....	5
Caveats.....	5
<b>IBM N series simplifies SQL server data management</b> .....	<b>5</b>
High availability .....	5
Improved performance and asset utilization .....	5
Accelerated test and development with rapid cloning .....	6
Lightening fast backup and restore with SMSQL 2.0 .....	6
SMSQL benefits at a glance.....	6
SMSQL 2.0 enhancements .....	6
Best practice software solutions for managing SQL server environments .....	6
<b>Database and storage logical and physical layout</b> .....	<b>7</b>
Mapping databases to LUNs .....	7
How to use the Information in Table 2.....	7
System databases and files .....	8
External SQL Server files .....	8
Mapping LUNs to IBM N series volumes .....	8
The foundation: Large aggregates .....	8
Recommended approach for “most” SQL Server deployments .....	9
Volume layout considerations .....	10
Managing space.....	11
Snapshot copies .....	11
Space reservation.....	12
Disk space consumed by read/write snapshot copies .....	12
Space requirements overview .....	12
<b>Backing up and restoring databases with SMSQL</b> .....	<b>13</b>
Backup recommendations .....	13
Archiving backups.....	14
<b>Host and storage system installation and configuration best practices</b> .....	<b>15</b>
IBM N series Storage System requirements.....	15
Installing and Configuring SnapDrive 3.1.1R1 (or higher) .....	15
Preparing for installation.....	15
SnapDrive 3.1.1R1 or higher installation tips .....	16
Creating and managing LUNs with the SnapDrive GUI .....	16
Installing and configuring IBM N series SMSQL.....	16
General information .....	16
SQL Server 2000/2005 resource use and concurrent database backup .....	16
Database consistency check.....	17
SMSQL 2.0 in a Microsoft Cluster Services (MSCS) environment .....	18



Snapshot writable LUNs.....	19
Storage mirroring and DR.....	19
Mirroring frequency.....	19
Key SnapMirror concepts and tips .....	20
Checklist for configuring SnapManager and SnapMirror .....	20
<b>Best practices for ongoing management and monitoring .....</b>	<b>22</b>
Snapshot reserve usage monitoring .....	22
Disk space usage monitoring.....	23
IBM N series storage system event monitoring .....	23
Terminal server or remote desktop.....	23
<b>Summary .....</b>	<b>24</b>
<b>Appendix A: SMSQL 2.0 best practices recap.....</b>	<b>25</b>
<b>Appendix B: Consolidation methodologies.....</b>	<b>27</b>
Phase I: Analysis of existing SQL Server instances.....	27
Phase II: Designing and migrating to the new environment .....	29
Consolidation results.....	30
Key observations:.....	31
<b>Appendix C: Sample SQL Server sizing script.....</b>	<b>32</b>
<b>Trademarks and special notices.....</b>	<b>34</b>



## Abstract

---

*Increasingly, organizations select Microsoft SQL Server to provide a backend data store for mission-critical, line-of-business applications. Microsoft SQL Server 2005 offers significant architectural enhancements in performance, scalability, availability, and security. The success of deployment hinges on making the right choices and configurations upfront. This report offers best practices for deploying and using IBM System Storage N series with SnapManager for SQL Server 2.0 with Microsoft SQL Server 2000 and 2005 and supporting software.*

## Introduction

---

Nearly all of today's business applications are data-centric, requiring fast and reliable access to intelligent information architectures that can often be provided only by a high-performance relational database system. Over the last decade, many organizations have selected Microsoft® SQL Server to provide a backend data store for mission-critical, line-of-business applications. Microsoft SQL Server 2005 offers significant architectural enhancements in performance, scalability, availability, and security. As such, SQL Server environments continue to proliferate and become more complex.

It is disruptive and costly when customers, employees, partners, and other stakeholders are adversely affected by database outages or performance-related impediments. For many systems, the underlying physical storage architectures supporting SQL Server are often expected to scale in a linear capacity, but frequently bottlenecks arise and backup processes get slower as databases get larger. SQL Server databases are growing significantly larger to meet the needs of today's organizations, while business requirements dictate shorter and shorter backup and restore windows. It is not an unusual occurrence for organizations to take days to restore a large SQL Server database, presuming all the required backup media are close at hand and error-free. For many organizations this can have a direct and devastating impact on revenue and customer good will. It is clear that conventional backup mechanisms do not scale and many organizations seek new, innovative solutions to solve increasingly stringent business needs.

### Purpose and scope

The success of any software or infrastructure and software deployment hinges on making the right choices and configurations upfront. This report provides best practices for deploying and using IBM® System Storage™ N series with SnapManager® for SQL Server 2.0 (SMSQL) with Microsoft SQL Server 2000 and 2005 and supporting software. Businesses that need to get the most from their IBM N series storage investment with SQL Server will benefit from putting this report's recommendations into practice. At a high level, the following areas will be covered:

- New features of SMSQL 2.0 and complimentary IBM N series technologies
- Best practices for both database and storage logical and physical layout with SQL Server and IBM N series storage systems and software
- Backup and recovery (BR) and disaster recovery (DR) considerations and tactics
- Best practices for installation, configuration, and ongoing use of IBM System Storage N series with SnapDrive® and SMSQL 2.0
- Ongoing management and monitoring best practices

**Note:** This document highlights specific best practices in boxed sections throughout. All best practices are also available in Appendix A.



## Intended audience

This report is intended for experienced SQL Server administrators, IT managers, and storage administrators who have reviewed the documents below:

- IBM N series with SnapDrive v.3.1.1R1 (or higher) installation/administration guide
- SMSQL 2.0 (SnapManager for SQL Server v 2.0) installation/administration guide
- IBM System Storage N series with Data ONTAP<sup>®</sup> 7.1 or later administration guide.

Readers should ideally have a solid understanding of the SQL Server storage architecture and administration as well as SQL Server backup and restore concepts. For more information about Microsoft SQL Server architecture, please refer to SQL Server Books Online (BOL).

## Caveats

- Applications utilizing SQL Server as a back end may have specific requirements dictated by the design characteristics of the application and are beyond the scope of this technical report. For application-specific guidelines and recommendations as they relate to physical and logical database layout, please contact your application provider.
- Best practices for managing SQL Server environments will focus on IBM N series storage operating system, Data ONTAP 7.1, the version at the time of report writing.

## IBM N series simplifies SQL server data management

---

### High availability

IBM N series storage systems are uniquely scalable and highly available with the ability to scale from 720GB all the way up to half a petabyte. Fabric-attached storage (FAS) systems offer a rich set of features with inherent redundancy of robust hardware components. Clustered FAS systems are available for even higher levels of availability. Innovations like double parity (IBM System Storage N series with RAID-DP<sup>™</sup>) enhance availability by protecting against secondary drive failures that may occur during RAID (redundant array of inexpensive disks) reconstructs. The SnapDrive family of products offers Multi-Path I/O (MPIO) solutions for both iSCSI and Fibre Channel (FC) protocols, providing redundant paths from hosts to FAS systems. For more information about MPIO, please visit the [Microsoft MPIO site](#).

### Improved performance and asset utilization

IBM System Storage N series with FlexVol<sup>™</sup> technology introduced in Data ONTAP 7.1 makes it possible to optimize storage utilization by logically partitioning IBM N series storage (aggregates) into smaller virtualized volumes (FlexVol volumes). Many SQL Server environments support small databases with very high I/O appetites, often requiring many more disk drives for throughput than required for space. FlexVol makes it possible to create a base aggregate container with many disk drives and logically partition the aggregate into smaller virtualized volume structures. By combining the same number of spindles previously sliced up across many conventional volumes, much higher throughput and storage asset utilization can be achieved while still maintaining the benefits of discreet dedicated volumes. FlexVol volumes can be created, expanded or reduced in mere seconds, regardless of size.



## Accelerated test and development with rapid cloning

An IBM System Storage N series with Snapshot™ copy of a database in a FlexVol volume can easily be cloned for testing or development purposes.

## Lightening fast backup and restore with SMSQL 2.0

IBM N series with SMSQL 2.0 is an integral part of the complete IBM N series data management solution that integrates Microsoft SQL Server with renowned features intrinsic to IBM N series storage technology.

### SMSQL benefits at a glance

- Near instantaneous backup and fast restore of entire SQL Server databases and full text indexes (SQL Server 2005) using IBM N series with Snapshot technology
- Easy migration wizards to move databases to storage-area network (SAN) and IP SAN environments
- Easy-to-use, intuitive graphical user interface (GUI)
- Rich backup scheduling and reporting
- Integration with IBM System Storage N series with SnapMirror® for wide-area network (WAN) data replication.

### SMSQL 2.0 enhancements

SMSQL 2.0 extends the features and functionality of the previous version SMSQL 1.5, in the following ways:

- Full support and integration with Microsoft SQL Server 2005
- BR of Microsoft SQL Server 2005 full text indexes
- Ability to restore clustered databases without taking the virtual server offline (SnapDrive 3.2 only)
- New options, including:
  - Option to skip transaction log backup prior to a database restore operation
  - Option to limit IBM System Storage N series with AutoSupport notifications to SnapManager failure events
  - Option to run "UPDATE STATISTICS" on tables before detaching the databases.

## Best practice software solutions for managing SQL server environments

For optimal management of SQL Server environments, it is recommended that you utilize the comprehensive set of solutions outlined below.

Solution	Benefit
Data ONTAP 7.1 or higher	Higher storage asset utilization, performance, and overall flexibility
SnapManager for SQL Server 2.0	Near instantaneous backup and fast restore
SnapDrive 3.2	For MPIO support with iSCSI and FCP
SnapMirror	Simplified disaster recovery
IBM System Storage N series with FlexClone™	Accelerated development, test, and change management

Table 1. Best-practice SQL Server management.



We will now focus on key practical areas related to best-practice deployments and configurations with SQL Server and IBM N series storage and software, starting with database and storage physical layout.

## Database and storage logical and physical layout

This section discusses and makes recommendations for optimal layout of SQL Server databases with IBM N series storage systems using SMSQL 2.0 for BR. Layout of several logical entities, such as IBM N series logical unit numbers (LUNs) and FlexVol volumes are discussed, as well as physical entities such as SQL Server database files and aggregates.

### Mapping databases to LUNs

Table 2 provides an overview of recommended physical LUN layout for various databases based on physical size as well as SMSQL requirements.

LUN Mapping to SQL Server Database Size				
	Small Database	<sup>1</sup> Many Small Databases < 2TB	<sup>2</sup> VLDB > 2TB	VLDB > 4TB
Single LUN	X	X		
Two LUNs	X	X		
Three or more LUNs			X	X
The LUN can be shared by many databases	X	X		
The LUNs cannot be shared by any other database			X	X
Multiple data files in a SQL Server file group			X	X

Table 2. Database setup and recommendations overview.

### How to use the Information in Table 2

Placement of SQL Server files on LUNs primarily depends on the number of databases deployed per server, the total size of the database today, and projected growth. Determining the placement of database files depends on several related factors, but there are some rules to follow:

- Several databases can share LUNs when all database files are in one or two LUNs only:
  - **Scenario I:** A LUN has been mounted on 'X,' and all files (log, full text catalog files, and data files) belonging to MyDB are placed in 'X,' in which case all files belonging to MyDB2 can also be located on 'X' .
  - **Scenario II:** A LUN has been mounted on 'D,' and another LUN has been mounted on 'L.' MyDB's transaction log file is placed in 'L,' and MyDB's data files are placed in 'D,' in

<sup>1</sup> The term “many” is meant to depict more than two databases.

<sup>2</sup> VLDB stands for “very large database.”



which case MyDB2's transaction log and full text catalogs can also be placed in 'L,' and its data files can be placed in 'D'.

- **When files belonging to a database are placed in more than two LUNs, no other database can use any of the LUNs.** This includes placement of the transaction log files, full text catalogs (SQL 2005 only) and data files. For example, if three LUNs are mounted on 'X,' 'Y,' and 'Z,' and MyDB has its log file located in 'X,' some data files in 'Y,' and full text catalog files in 'Z,' then 'X,' 'Y,' and 'Z' cannot be used by any other database.
- **System databases should be placed on their own LUN.** When system databases are combined with user databases, all user database backups revert to a standard stream backup and snapshot copies are not used. There are also situations where it is beneficial to isolate the system database LUN on its own FlexVol volume. This is discussed later.
- **Very large databases (VLDBs): If a single database is greater than 2TB or has been forecasted to grow beyond 2TB, multiple data files (.ndf files) in at least one or more file group should be added and distributed evenly across multiple LUNs.** SnapDrive 3.X will restrict the largest LUN size to 2TB on both Windows® 2000 and Windows 2003 (later versions may not; however this paper focuses on versions at the time of writing). Thus, if the database that is being migrated to the IBM N series storage system is going to be greater than 2TB, it is recommended that the base file group be deployed with several data files, each distributed across three or more LUNs respectively.

**Note:** In clustered environments, remember that the total number of available LUNs (drive letters) is equal to that of one node in multi-node configurations.

## System databases and files

### External SQL Server files

SQL Server deployments may use SQL Server Integration Services (SSIS), Data Transformation Services (DTS) packages, or externally defined .NET assemblies and functions. SMSQL will not back these files up, so other means should be put in place to ensure data protection.

## Mapping LUNs to IBM N series volumes

### The foundation: Large aggregates

There are two prevailing models for constructing and laying out IBM N series storage for SQL Server storage: shared and dedicated. The shared model provides many more spindles for greater use by aggregating all available disks into a single resource available to all database instances. This approach essentially translates into using all spindles to create one or two large Data

**Best Practice:** Always place system databases on a dedicated LUN and FlexVol volume.

If user databases are combined with system databases on the same LUN, sSnapshot backups will not work and SnapManager will revert to the conventional stream-based backup, dumping all data, "block for block" to the snapinfo LUN.

Keeping system databases on a dedicated LUN and FlexVol volume separate from user databases has the added benefit of keeping deltas between sSnapshot copies smaller (as they do not hold changes to tempdb). Tempdb is recreated each time SQL Server is restarted and provides a scratch pad space where SQL Server manages the creation of indexes and various sorting activities. The data in tempdb is thus transient and does not require backup support. Since the size and data associated with tempdb can become very large and volatile in some instances, it should be located on a dedicated FlexVol volume with other system databases on which sSnapshot copies do not take place. This also preserves bandwidth and space when SnapMirror is used.





ONTAP 7.1 aggregates. The large aggregate is then used as a central “pool of storage” on which FlexVol volumes can be provisioned, expanded, or contracted on demand. The benefit of this type of deployment is that space utilization can be optimized while throughput is shared across a greater number of spindles than previously possible.

In contrast, the dedicated model seeks to break the same pool of disks into much smaller discreet physical volumes for exclusive use and management by each server. The drawback to this approach is twofold: (1) over-provisioning of space is more difficult to overcome, and (2) throughput of the total storage array investment is adversely affected if one volume is I/O constrained and unable to use other dedicated disk resources in the array that may, at the same time, remain idle.

In one study, using database workloads in shared pools of storage provided significant performance benefits over dedicated disk deployments. This study shows great performance gains for the shared model in two core areas:

- Mixing decision support system (DSS) and online transaction processing (OLTP) workloads vs. separation of workload types
- Combining log and data files vs. separation of log and data files.

### Recommended approach for “most” SQL Server deployments

In general, for most deployments, organizations looking to get the most out their IBM N series storage investment should strongly consider taking one of two approaches. The first prescribes the creation of a single foundational aggregate for all SQL Server databases, consisting of RAID-DP groups composed of a sweet spot configuration of 16 spindles each. This approach can provide large performance benefits for all databases leveraging the single aggregate. Like most designs there are pros and cons beyond simply performance that should be considered. While a single aggregate can provide a very high level of availability, an outage can occur in situations where one RAID-DP group loses more than two disks. While this is unlikely, there are scenarios in which the single aggregate approach does not provide the right level of database availability.

Another approach to provide an additional layer of protection from data loss would be to use the same number of disks split into two aggregates—one housing the transaction logs and the other the data files. In the event of an outage of the data volume, the transaction log could still be backed up for recovery. Consider this deployment where recoverability requirements go beyond double disk failure.

Finally, for environments requiring unusually high availability, consider using IBM System Storage N series with SyncMirror®. SyncMirror is a solution that provides complete local data mirroring of an aggregate and can be deployed for an even higher level of protection when necessary.

To get the most out of large aggregate deployments, please follow these guidelines:

The number and type of disks in the aggregate depend on the workload characteristics of all databases being deployed to the aggregate.

#### **Best Practice: Large aggregate key points (settings and guidelines):**

Turn volume setting `minra` on for aggregate and all FlexVol volumes.

## Volume layout considerations

Once the Data ONTAP 7.1 aggregate layout model has been selected and implemented, FlexVol volumes and LUN objects will need to be created for each SQL Server instance. This section will focus around how to most effectively lay out FlexVol volumes with SQL Server databases.

Keep in mind FlexVol volumes provide a boundary for:

- Snapshot copies and log backup schedules
- Mirroring schedules

**Snapshot:** Because SnapManager leverages volume-level snapshot copies, all user databases residing in a given volume will be backed up by the snapshot copy. This becomes apparent in the SMSQL GUI when attempting to back up a single database sharing a volume with other databases. All databases residing in the volume get selected for backup regardless of unique backup requirements and cannot be overridden. The only exception to this is the system databases, which are handled by streaming backup, but even they will be included in the snapshot backup when sharing a volume with user databases.

Transaction log backup settings are also specific to the entire volume, so if you just want to dump the log for say, one or two databases in a volume hosting numerous databases, you will be forced to back up all transaction logs for all databases. Please review the best practice guidelines below.

**Mirroring:** Many organizations have requirements for mirroring of SQL Server databases to geographically dispersed locations for data protection and business continuance purposes. Like Snapshot, SnapMirror works at the volume level, so proper grouping of databases by the high-availability index (discussed in further detail in Appendix B: Consolidation Methodologies) can be particularly useful. Locating one database per FlexVol volume provides the most granular control over SnapMirror frequency as well as the most efficient use of bandwidth between SnapMirror source and destination. This is because each database is sent as an independent unit without the additional baggage of other files such as `tempdb` or databases that may

### Best Practice: Use FlexVol volumes when more granular control is required.

When backup requirements are unique to each database, there are two possible paths that can help optimize snapshots and transaction log backups:

- **Most Granular Control:** Create separate FlexVol volumes for each database for more granular control of backup schedules and policies. The only drawback to this approach is a noticeable increase in administration and setup.
- **Second Most Granular:** Try to group databases with similar BR requirements on the same FlexVol volume. This can often provide similar benefits with less administration overhead.

### Best Practice: Avoid sharing volumes between hosts.

It is not recommended to let more than a single Windows server create and manage LUNs on the same volume. The main reason is that snapshot copies are based at the volume level and issues can arise with snapshot consistency between servers. Sharing volumes also increases chances for busy snapshot copies.

not share the same requirements. While sharing of LUNs and volumes across databases is sometimes required because of drive letter limitations, every effort should be made to group databases of like requirements when specific mirroring and backup policies are implemented.

Performance: In some circumstances, even in large aggregate environments, it can be beneficial to separate transaction log files from data files into different FlexVol volumes. This is because the IBM System Storage N series with WAFL<sup>®</sup> file system has optimizations that seek to increase locality of data on the aggregate at the FlexVol layer. At a high level, here is the sequence of events that take place during a write operation:

1. Data is written to the battery-backed non-volatile random access memory (NVRAM) portion of the IBM N series storage system and acknowledged immediately to the SQL Server host as being written.
2. Data ONTAP 7.1 then logically sorts all writes into chunks determined by each FlexVol volume.
3. A consistency point (CP)<sup>3</sup> takes place, during which time blocks are written to the next free blocks on the disk in a contiguous manner for each target FlexVol volume.

**Best Practice: For increased SQL Server performance, use separate FlexVol volumes for transaction log and data files.**

In environments where performance of databases is especially important, consider separating transaction log and data files into separate FlexVol volumes. Keep in mind that even though the data may actually reside in some cases, on the same physical aggregate, better locality of data can be achieved by using FlexVol volumes for logical separation of data.

**Note:** Data ONTAP 7.1 can support up to 200 FlexVol volumes per storage system and 100 for clustered configurations. Also, drive letter constraints in MSCS configurations may require combining log and data into a single LUN or FlexVol volume.

Hence, combining log and data files on the same FlexVol volume can cause the two types of data to become indiscriminately interspersed when written to the next free extents. When a sequential read is issued for committed data blocks, the storage system may have to skip over log records to find the needed data blocks, increasing the potential for additional I/O operations.

## Managing space

### Snapshot copies

It is beyond the scope of this paper to explore snapshot copies in great detail. However, it is necessary to understand the fundamentals of the unique features of IBM N series with Snapshot functionality and how it relates to Microsoft SQL Server 2000 and 2005.

IBM N series with Snapshot backups occur in a matter of seconds and typically each copy consumes only the amount of data that has changed since the last copy was created. Thus, snapshot copies

---

<sup>3</sup> A consistency point (CP) refers the instant at which the NVRAM portion of the NetApp storage system is flushed to disk.

consume minimal disk space while providing up to 255 online point-in-time images. The amount of disk space consumed by an individual snapshot copy is determined by two factors:

- The rate data changes within file systems (in MB/sec or MB/hour, for example)
- The amount of time that elapses between creation of snapshot copies.

The measure of change (in megabytes, gigabytes, etc.) that occurs in the volume between creations of snapshot copies is often referred to as the delta. The total amount of disk space required by a given snapshot copy equals the delta changes in the volume along with a small amount of snapshot metadata.

### Space reservation

Space reservation is designed to ensure that protected files (or files that have space reservation turned on) always have the free space they expect so that a snapshot copy of the LUN can complete even if 100% of the SQL Server databases blocks have changed. Data ONTAP 7.1 provides for space reservation at the FlexVol, qtree, and file levels. Any file may be designated as protected, and is always turned on by default for LUNs.

When creating a LUN, the IBM N series storage system verifies there is enough available disk space to accommodate the specified size. By default, WAFL reserves blocks equal to two times the specified size of the LUN (when IBM N series with Snapshot is used) so that all future writes can be completed. If space reservation was not enabled, it would be possible to oversubscribe the available storage. If this occurs, SQL Server will, because its space is preallocated, interpret it to mean the storage device is bad, and it will mark affected databases as suspect.

**Best Practice: Leave space reservation enabled.**

Also, to prevent issues surrounding data corruption, avoid changing the default fractional reserve<sup>1</sup> settings.

### Disk space consumed by read/write snapshot copies

Although read/write snapshot files are initially linked to the existing blocks in the snapshot copy, it is necessary for the IBM N series storage system to allocate blocks equal to the entire size of the database files should it be completely overwritten and a copy of it created. There must be enough available disk space to accommodate the entire size of the original LUN while the read/write snapshot copy is mounted. The space consumed by the read/write snapshot copy is marked as free disk space when it is dismantled using the Disconnect Disk command in SnapDrive. If the space is not available, consider using FlexClone to access snapshot copies of databases.

### Space requirements overview

Estimating storage requirements can be complicated, especially when a sizing project involves server consolidation. The basic areas of focus consist of several related objects:

1. System databases
  - How large does `tempdb` grow throughout the day?
2. User databases
  - How frequently will the database and log be backed up?
  - How large is each log dump?

- How large will the databases grow over the next two to three years?
- 3. SnapInfo
  - How many full backups will take place and be retained?
  - How many log backups?
  - How many stream-based backups of system databases?
- 4. Snapshot copy creation and retention
  - What is the approximate delta between snapshot copies?
  - How many transaction log and snapshot copies need to be available?
    - a. When and how often should an archive backup take place?

Work to establish a rough estimate for each of the four areas. Appendix B puts all guidelines to practice in a consolidation project. Please refer to this section for more detail.

## Backing up and restoring databases with SMSQL

---

How to back up and restore SQL Server databases is thoroughly discussed in the SnapManager for SQL Server 2.0 installation/administration guide. Only areas that have to be well understood will be reiterated here.

SMSQL will back up as follows:

- Snapshot copies will be created of all volumes used by the databases being backed up.
- System databases will be backed up using conventional streaming-based backup.
- User databases placed in a LUN that also contains system databases will be backed up using streaming-based backup.
- Backup of transaction logs will always use streaming-based backup to provide point-in-time restores.

SMSQL will restore database(s) by:

- SnapDrive will request a Single\_File\_SnapRestore<sup>4</sup> to restore all LUNs used by the database(s).
- Restoring selected databases out of many sharing one or two LUNs.
- Requesting SQL Server to restore the databases.
- Requesting SQL Server to apply selected backed-up transaction logs. This is configurable and can conduct either point-in-time or up-to-the-minute restores.
- The most efficient configuration is one user database per LUN.

### Backup recommendations

SMSQL 2.0's full backup using Snapshot is quick and does not adversely influence database performance compared to traditional backup with SQL Server BACKUP statements. Transaction log backup is functionally the same using SMSQL as the transaction log backup done using BACKUP Log statements and is streamed out to the SnapInfo directory for safekeeping. The length of time required to back up a transaction log depends mainly on the active section of the transaction log that has to be extracted and copied to the dump directory. When planning the backup and restore process, the

---

<sup>4</sup> Single\_File\_SnapRestore works by functionally copying only called upon files from within the specified Snapshot copy.



methodology or considerations are different from considerations when backup is done by Enterprise Manager or SQL Server Management Studio. The two main considerations are:

- Recovery point objective (RPO)—To what point in time does the data need to be recovered?
- Recovery time objective (RTO)—How long will it take to get the database back online and rolled forward or backward to the RPO?

Consider the hypothetical customer scenario below:

**Customer:** ACME Corp

**Database:** CommerceDB

**Failure Type:** Database corruption or deletion

**RTO:** It cannot take longer than 30 minutes to complete a restore

**RPO:** Data should be brought back to a time no later than 15 minutes of failure.

ACME's IT group has shown that a full backup every 30 minutes, with a transaction log backup every 15 minutes on the hour makes it possible to restore the database in 20 minutes (without the verification process) to the minimum recovery point objective of 15 minutes. Backup every 30 minutes during a 24-hour timeframe will create 48 snapshot copies and as many as 98 transaction log archives in the volume containing SnapInfo. An IBM N series volume can host up to a maximum of 255 snapshot copies so the proposed configuration could retain over five days of online snapshot data.

Thus, it is generally a good idea to categorize all your organization's systems that use SQL Server in terms of RTO and RPO requirements. Once these requirements have been established, SMSQL can be tailored to deliver the appropriate level of protection. Proper testing before deployment can also help take the guesswork out of what can be realistically achieved in production environments.

In situations where business requirements are not entirely clear, refer to the recommended default database backup frequency below.

#### **Best Practice: Minimum backup frequency:**

All BR processes should be created according to each environment's unique requirements, but when in doubt, the general recommendation is to back up the database every hour and the transaction log every 30 minutes during production time; possibly less often for off-peak hours.

## Archiving backups

It is strongly recommended to archive or mirror backup sets as soon as possible. Disasters do occur, and if the storage device containing the databases and backup snapshot images is adversely affected, it will not be possible to restore the databases from the primary storage system. Archive media can be tape, another FAS system, an IBM System Storage N series with NearStore<sup>®</sup> device or even local data drives available in the IBM System Storage N series. The archive process is described in further detail in SMSQL 2.0 installation/administration guide.

#### **Best Practice: Always archive or mirror snapshot copies.**

Archive or mirror backup sets as soon as possible.



## Host and storage system installation and configuration best practices

---

Several things need to be done in preparing an IBM N series storage system and SQL Server host to create a reliable system with optimal performance. Refer to the latest SnapDrive and SnapManager for SQL Server administration guides to ensure the proper licenses and options are enabled on the IBM N series storage device.

### IBM N series Storage System requirements

Minimum requirements for the IBM N series are:

- Data ONTAP 7.1 or later.
- SnapDrive 3.1.1R or higher. SnapDrive 3.2 preferred for cluster environments.
- Enable licenses for FC or iSCSI protocol, SnapManager, and SnapMirror according to requirements.
- At least one-gigabit card for connectivity (copper or fiber).

### Installing and Configuring SnapDrive 3.1.1R1 (or higher)

#### Preparing for installation

Prior to launching the installation of SnapDrive, use the checklist below to help eliminate the potential for errors or delays during or after the installation:

- Resolve any hardware, cabling, or network issues or other errors.
- Make sure all of the necessary software and printed or electronic documentation is organized and nearby before beginning.
- Configure domain name server (DNS), hostname, and IP address-related services
  - Verify that all related systems, including storage systems, servers, and clients, are configured to use an appropriately configured DNS server.
  - Manually add the IBM N series storage systems' host names to DNS.
  - Enable, configure, and test remote shell (RSH) access on the storage systems for administrative access (for more information, see the Data ONTAP 7.1 [or later] guide).
- License all of the necessary protocols and software on the storage system.
- Configure the storage systems to join the Windows 2000/2003 Active Directory domain by using the IBM System Storage N series with FilerView® administration tool or by running `filer> cifs setup` on the console (for more information, see the Data ONTAP 7.1 [or later] guide).
- Make sure the storage systems date and time are synchronized with the Active Directory domain controllers. This is necessary for Kerberos authentication. A time difference greater than five minutes will result in failed authentication attempts.
- Verify that all service packs and hotfixes are applied to the Microsoft Windows 2000 Server or Microsoft Windows 2003 Server and Microsoft SQL Server 2000 and 2005.

## SnapDrive 3.1.1R1 or higher installation tips

Select the same account used by the Microsoft SQL Server services when selecting the service account for the SnapDrive and SnapManager for Microsoft SQL Server services.

When creating or configuring the properties for the domain service account, select the “Password never expires” checkbox. Doing so protects the account and service from failing due to user password policies.

Reminder: It is important to make certain the service account has the following permissions:

- Read and write or full control access to the locations in which LUNs will be created (likely if it's already a member of the administrator's group).
- RSH access to the storage system's. For more information on configuring RSH, see the Data ONTAP administration guide.

### Best Practice: Tips for creating LUNs.

- **When specifying a UNC path to a share of a volume to create a LUN, use IP addresses instead of host names.** This is particularly important with iSCSI, as host-to-IP name resolution issues can interfere with the locating and mounting of iSCSI LUNs during the boot process.
- **Use SnapDrive to create LUNs** for use with Windows to avoid complexity.
- **Calculate disk space requirements** to accommodate for data growth, snapshot copies, and space reservations.
- **Leave automatic snapshot scheduling off** as configured by SnapDrive.

## Creating and managing LUNs with the SnapDrive GUI

Once installed, SnapDrive can be used to create LUNs on IBM N series storage systems for use by Windows 2000 and 2003 hosts.

## Installing and configuring IBM N series SMSQL

### General information

After following the steps outlined in the SnapDrive installation guide and listed in the previous sections of this paper, proceed with the installation of SMSQL. The SMSQL application is a powerful tool that orchestrates the migration of system and user databases as well as controlling backup schedules and managing snapshot copies.

### SQL Server 2000/2005 resource use and concurrent database backup

When SMSQL backs up one or more databases managed by SQL Server, SQL Server requires internal resources (worker threads) to manage each database that is being backed up. If not enough resources are available, the backup operation will fail. The SnapManager Configuration wizard gives the number of databases that can be backed up concurrently (see Figure 1).



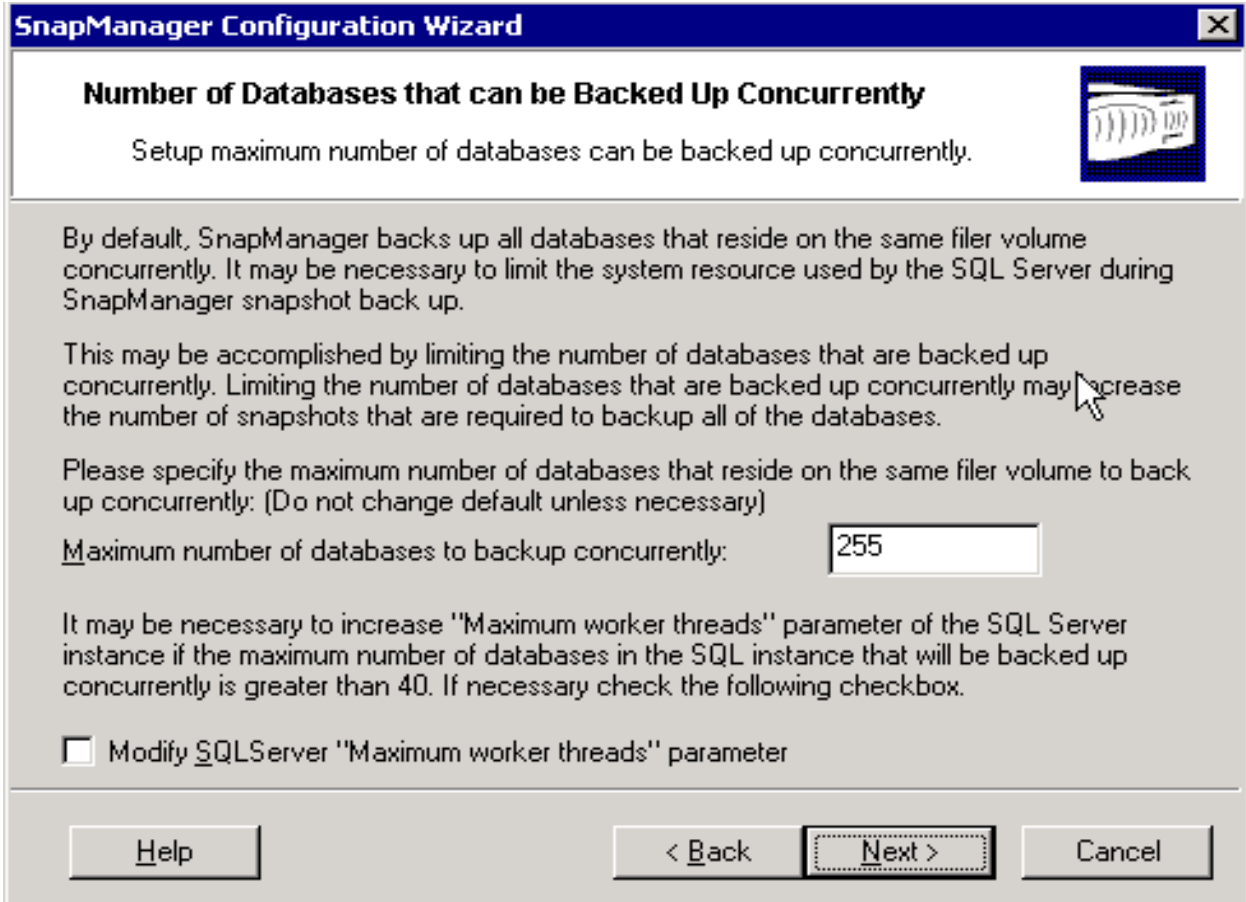


Figure 1. Concurrent backup screen.

As a general rule, leave the worker thread settings at the default settings. If you have more than 40 databases in the instance, consider selecting the *Modify SQLServer "Maximum worker threads" parameter* and increasing this value in the subsequent screen.

### Database consistency check

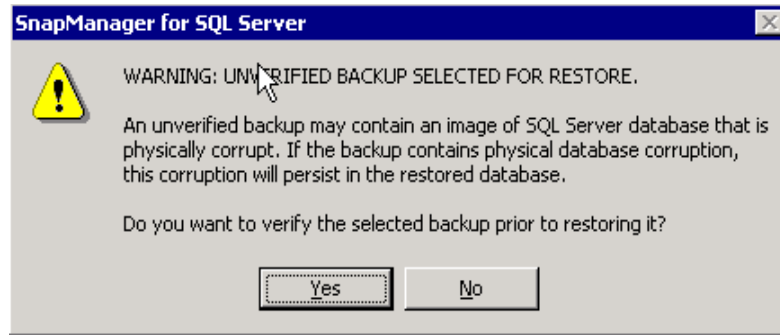
Databases can be so large that the verification process takes an almost unreasonable amount of time to complete. It is not always practical to check consistency every time the database is backed up, in which case it is recommended to check the consistency of the database once a week or more and to check the consistency of every transaction log backup.

Many database administrators (DBAs) have experienced how problematic and time-consuming it is to restore an inconsistent database. After much checking the DBA realizes the database was inconsistent before it was even backed up. Hence, while it is not always convenient to check the database's consistency frequently, it is strongly recommended. SMSQL will not let a DBA restore a backed-up database that has not been checked for inconsistencies by default, but it does provide the

**Best Practice: Run Consistency Checker (DBCC CHECKDB) to validate database before and after migration.**

For usual operations, also try to execute the verification process during off-peak hours to prevent excessive load on the production system. If possible, offload the entire verification process to another SQL Server instance running on another server.

option to let a database be restored, even when consistency has not been verified. The DBA has to acknowledge, during the restore process, that the safety net will be bypassed, as shown in Figure 2.



**Figure 2. Bypassing database verification.**

Verification of backed up databases must be done by a nonclustered SQL Server instance either installed on the local Windows server or on another Windows server that can be a dedicated verification server. The reason for this is simply that a virtual SQL Server cannot be used for verifying a database's consistency. The LUN in the snapshot copy will be restored during the verification process, and the temporary restored LUN cannot be a cluster resource and therefore cannot be included in SQL Server's dependency list. A virtual SQL Server instance can be used for verifying the online database.

### **SMSQL 2.0 in a Microsoft Cluster Services (MSCS) environment**

When configuring a Microsoft cluster environment, begin the setup by following the instructions in the SnapDrive installation/administration guide. Make certain the cluster functions correctly by using the Cluster Manager to move cluster groups from node to node. After verifying the move group works correctly, proceed to install SQL Server and then SMSQL. The result will be one cluster group per virtual SQL Server instance. Each cluster group will contain at least three LUNs: one for system databases, one for user-created databases, and one for SnapInfo.

For each virtual SQL Server follow these steps:

1. From SnapDrive, create a cluster-aware LUN that is located in the cluster group that will contain SQL Server—for example, SQL Group 1 (a new cluster group can be created from SnapDrive or from the Cluster Manager).
2. Create a second cluster-aware LUN for user/application-created databases in cluster group SQL Group 1.
3. Create a third cluster-aware LUN for SnapInfo in cluster group SQL Group 1.
4. Install the virtual server in SQL Group 1 using the LUN for system databases.
5. Make the LUN for user databases a resource for the virtual server. From Cluster Manager:
  - a. Open SQL Group 1.
  - b. Right-click the virtual SQL Server.
  - c. Select Properties.
  - d. Select Dependency.
  - e. Make the LUN appear in the dependency list.
  - f. Make sure that the LUN is available by creating a dummy database from the SQL Server Enterprise Manager (delete the database).

## Snapshot writable LUNs

Snapshot backups of SQL Server databases are read-only point-in-time images. This protects the integrity of the snapshot backups. In certain situations, SMSQL will restore a LUN in a snapshot copy for temporary read/write access (restored to alternative location using a writable snapshot<sup>5</sup>). Writable snapshot copies are used during a verification process, when a DBA restores a database to an alternative location for recovering from operational mistakes or in some cases when external archive backups take place. A restored writable snapshot copy is for short-term use only.

**Best Practice: Avoid creating snapshot copies on volumes with active snapshot writable<sup>1</sup> LUNs.**

Creating a snapshot copy of a volume that contains an active snapshot copy is not recommended because this will create a “busy snapshot”<sup>1</sup> issue.

Avoid scheduling SMSQL snapshot copies when:

- Database verifications are running
- Archiving a snapshot backed LUN to tape or other media.

## Storage mirroring and DR

IBM N series with SnapMirror technology mirrors data to one or more IBM N series storage volumes over a local area network (LAN) or WAN to another IBM N series storage system. Once source and destination relationships are established, a SnapMirror baseline transfer initializes the mirror to create a replica of the source on the destination. SnapMirror maintains the synchronization of the replica through efficient, incremental updates that use bandwidth intelligently to distribute deltas across a WAN.

### Mirroring frequency

The frequency of SnapMirror update events (in most environments) is most often determined by the frequency of SMSQL backups. SnapDrive triggers SnapMirror updates upon completion of a SnapManager backup procedure. A supplemental type of snapshot copy optimized for use with SnapMirror called “rolling snapshots” can be used to

augment standard SQL Server backups. Rolling snapshot copies are controlled outside of SMSQL by using the scripting executable, `sdcli.exe` for SnapDrive. Be mindful of some rolling snapshot best practices discussed below.

**Best Practice: When using supplemental rolling snapshot copies:**

- Be sure to thoroughly read the SMSQL installation and configuration guide section relating to minimizing data loss exposure.
- Be sure to configure rolling snapshots to occur only between SMSQL backup and SnapMirror processes to prevent snapshot schedule overlap.

---

<sup>5</sup> A writable Snapshot or snapshot writable LUN is a LUN that has been restored from a Snapshot copy and enabled for writes. A file with an `.rws` extension holds all writes to the snapshot copy.



## Key SnapMirror concepts and tips

- SnapMirror relationships are based on sources and destinations.
- A destination updates its mirrored copy or a replica of its source by "pulling" data across a LAN or WAN when an update event is triggered by the source. The pull events are triggered and controlled by SnapDrive.
- Consistent with the pulling nature of SnapMirror, relationships are defined by specifying sources from which the destination IBM N series storage systems synchronize their replicas.
- Destination systems are identified on source systems by assigning destinations privileges via the `snapmirror.access` protocol access option or by inclusion in the `snapmirror.allow` file.
- SnapDrive 3.x works with volume SnapMirror (VSM) only. Qtree SnapMirror (QSM) is not supported.

As discussed earlier, SnapManager is integrated with SnapDrive, which interfaces directly with an IBM N series storage system by using either iSCSI or FC disk-access protocols. SnapManager relies on SnapDrive for disk management, controlling snapshot, and SnapMirror events. Table 3 illustrates the integration of SMSQL, SQL Server 2000/2005, SnapDrive, and IBM N series storage devices.

SnapManager for SQL Server 2.0	
SnapDrive 3.x	SQL Server (VDI)
Online Storage	SnapMirror

Table 3. SMSQL and SnapMirror.

## Checklist for configuring SnapManager and SnapMirror

1. Install (via common internet file system, CIFS, setup) both storage systems into the same Windows domain as SQL Server.
2. Configure the SQL Server, SnapDrive, and SnapManager Win32 services to use the same logon service account.
3. Make sure the SnapDrive service account has the "Log on as a service" rights in the Windows 2000\2003 operating system (normally occurs during installation).
4. Verify that RSH commands work between the SQL Servers and both storage devices using the specified service account.
5. License and enable FC or iSCSI protocol on both storage devices.
6. License and enable SnapMirror on both storage devices.
7. Establish SnapMirror relationships.
8. Make sure the storage device's SnapMirror schedule is turned off by assigning the "- - - -" value (four dashes separated by spaces) in the `custom schedule` field.
9. Initialize the mirror configurations.
10. SnapMirror updates to the specified destinations will occur after the completion of every SMSQL backup.
11. Use SnapDrive and FilerView to verify the successful completion and state of the configured mirrors.

Virtual Disk ID [Bus,Target...	Disk Location	Filer	File Path	Protocol Type	Drive Letter	Disk Size	SnapMirrored
VirtualDisk{0,1,0} (F)	\\eddie\exchhq3\exchdisk3.lun	eddie	\\vol\exchhq3\exchdisk3.lun	LUN (FCP)	F	1004 MB	Yes
VirtualDisk{0,1,1} (G)	\\eddie\exchhq4\exchdisk4.lun	eddie	\\vol\exchhq4\exchdisk4.lun	LUN (FCP)	G	1004 MB	Yes
VirtualDisk{0,1,2} (H)	\\eddie\exch1\exchdisk1.lun	eddie	\\vol\exchhq1\exch1\exchdisk1.lun	LUN (FCP)	H	1004 MB	No
VirtualDisk{0,1,3} (I)	\\eddie\exch2\exchdisk2.lun	eddie	\\vol\exchhq2\exch2\exchdisk2.lun	LUN (FCP)	I	1004 MB	No

Figure 3. SnapMirror status in SnapDrive.

FileView for Sammy - Microsoft Internet Explorer

### SnapMirror Report

SnapMirror → Report

Source (filer:location)		Destination (filer:location)				
eddie:exchhq3		sammy:exchhq3				
Current:	Status	Size (KB)	State	Content State	Lag	Type
	Idle	0	Snapmirrored	Replica	01:16:48	-
Last Transfer:	Size	Duration	Type	Base Snapshot	Base Timestamp	
	336 KB	00:00:04	Update	eddie(0033580113)_exchhq3.3	Thu Jan 30 14:09:40 PST 2003	

Source (filer:location)		Destination (filer:location)				
eddie:exchhq4		sammy:exchhq4				
Current:	Status	Size (KB)	State	Content State	Lag	Type
	Idle	0	Snapmirrored	Replica	01:16:41	-
Last Transfer:	Size	Duration	Type	Base Snapshot	Base Timestamp	
	15.6 MB	00:00:05	Update	eddie(0033580113)_exchhq4.3	Thu Jan 30 14:09:56 PST 2003	

Figure 4. SnapMirror status in FileView.

## Best practices for ongoing management and monitoring

### Snapshot reserve usage monitoring

The task of monitoring the snapshot reserve is automatically configured at the time of LUN creation. Simply monitor the application event log for warning events generated in the SnapDrive source and snapshot Monitor event categories. Figure 5 demonstrates that due to snapshot consumption, the reserve must be expanded to 63%, and there is not enough disk space available to do so. To rectify this situation, simply grow the FlexVol volume, remove some of the oldest SMSQL snapshot copies, or add new disk resources to the aggregate.

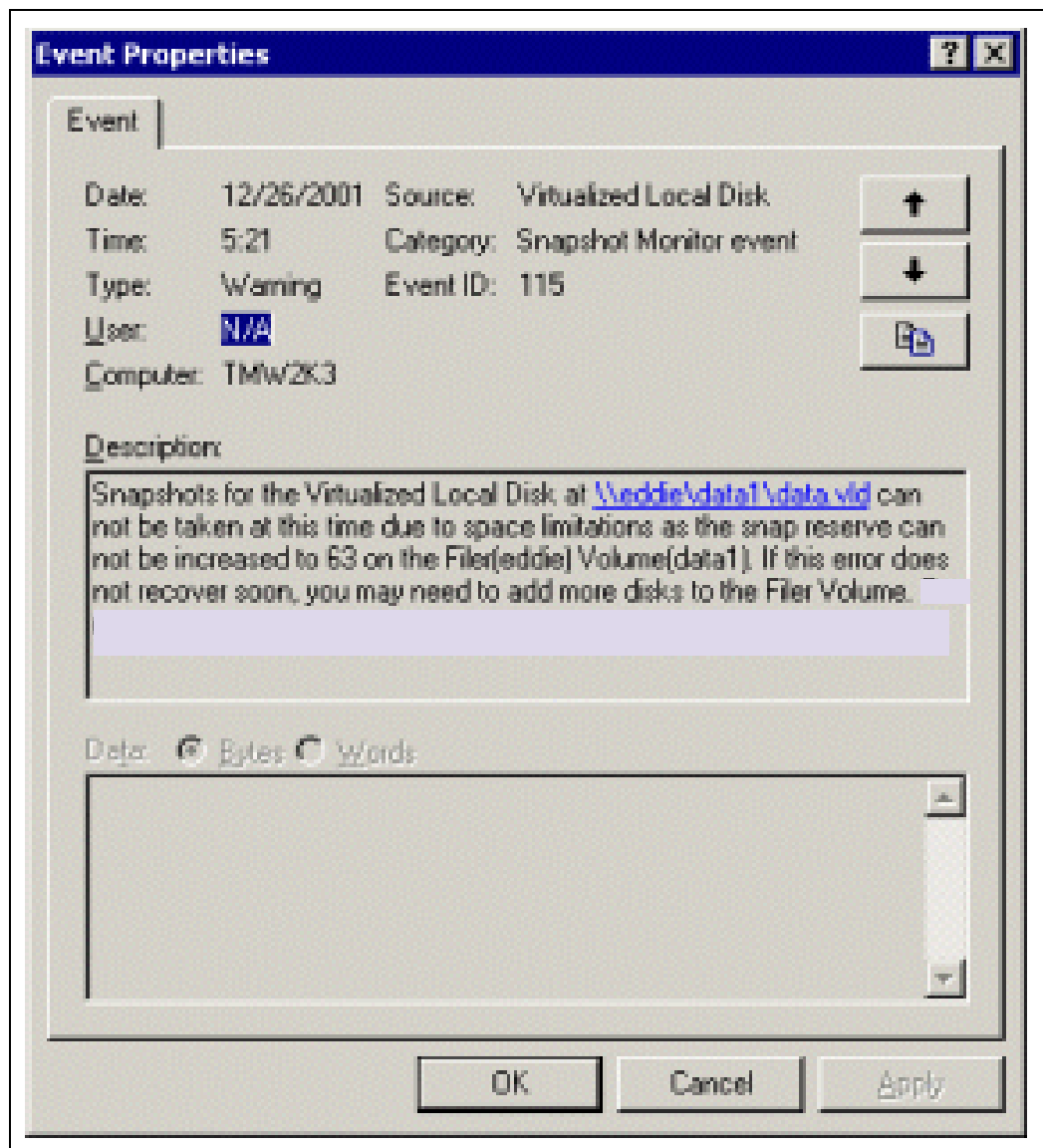


Figure 5. Space alert.



## Disk space usage monitoring

The amount of disk space used by each database managed by SQL Server 2000 and 2005 should be monitored to ensure that the LUNs do not run out of space. The default settings for SQL Server databases will automatically extend its preallocated file space when there is not enough free file space to extend a table (if the database has been defined with `auto extend on`). SQL Server will stop processing transactions when it cannot extend its file space.

Monitoring free space has to be done on two levels. The first level is to free preallocated space that is available for table extents. If Perfmon or a similar monitoring tool like Microsoft Operations Manager (MOM) is already used for monitoring the system performance, it is easy to add counters and rules to monitor database space. A database's properties will show its current size and how much space is free. When free space gets below or close to 64KB, the preallocated file space will be extended. The second level to monitor is free LUN space, which can be done with Windows Explorer. You may also use the stored procedure `sp_databases` to show the database files and sizes.

## IBM N series storage system event monitoring

Monitor the storage system's event logs to ensure proper operation. Issues may be discovered in the event logs that require administrative action. One such action may be to replace a disk in a volume if spare disks are not available. This task can be completed by using the FilerView utility built into Data ONTAP. Start the utility by pointing a Web browser to `http://filename/na_admin`. Next, click the FilerView link. FilerView will start and ask for the credentials of a storage system administrator. Once logged on to FilerView, click the Filer menu option on the left side of the screen. Then choose the Syslog Messages menu option. Review the log on the right side of the screen for any issues that may require administrative action. For more information, refer to the Data ONTAP administration guide.

## Terminal server or remote desktop

Microsoft Terminal Server for Windows is not recommended as a way of administering SnapDrive or SnapManager for SQL Server. When you create LUNs from a terminal server session, the drives can look like they were not created or have been disconnected when in fact they are functioning properly.

### **Best Practice: Use alternative means for remote management.**

It is recommended that you avoid Terminal Server for server management when possible. When using Terminal Server with Windows 2003 (not Windows 2000), you may use a remote desktop to connect if you use a command line parameter/console, shown below:

```
%SystemRoot%\System32\mstsc.exe /console
```

When using Windows 2000, it is advisable to use NetMeeting remote control.



## Summary

---

By providing the rich backup and restore features of SMSQL 2.0, performance and utilization optimizations of FlexVol, and robust data protection capabilities of SnapMirror, IBM N series provides a complete data management solution for SQL Server environments. The recommendations and examples presented in this document will help organizations get the most out of IBM N series storage and software for SQL Server deployments.



## Appendix A: SMSQL 2.0 best practices recap

---

**Best Practice:** Always place system databases on a dedicated LUN and FlexVol volume.

If user databases are combined with system databases on the same LUN, snapshot backups will not work and SnapManager will revert to the conventional stream-based backup, dumping all data, “block for block” to the snapinfo LUN.

Keeping system databases on a dedicated LUN and FlexVol volume separated from your user databases has the added benefit of keeping deltas between snapshot copies smaller as they do not hold changes to `tempdb`. `Tempdb` is recreated every time SQL Server is restarted and provides a scratch pad space in which SQL Server manages the creation of indexes and various sorting activities. The data in `tempdb` is thus transient and does not require backup support. Since the size and data associated with `tempdb` can become very large and volatile in some instances, it should be located on a dedicated FlexVol volume with other system databases on which snapshot copies do not take place. This will also help preserve bandwidth and space when SnapMirror is used.

**Best Practice:** Large aggregate key points (settings and guidelines):

- Turn volume setting `minra` on for aggregate and all FlexVol volumes.

**Best Practice:** For increased SQL Server performance, use separate FlexVol volumes for transaction log and data files.

In environments where performance of databases is especially important, consider separating transaction log and data files into separate FlexVol volumes. Keep in mind that even though the data may actually reside in some cases, on the same physical aggregate, better locality of data can be achieved by using FlexVol volumes for logical separation of data.

**Note:** Data ONTAP 7.1 can support up to 200 FlexVol volumes per storage system and 100 for clustered configurations. Also, drive letter constraints in MSCS configurations may require combining log and data into a single LUN or FlexVol volume.

**Best Practice:** Leave space reservation enabled.

To prevent issues surrounding data corruption, avoid using fractional reserve. Also leave space reservation on to avoid corruption.

**Best Practice:** Minimum backup frequency:

All BR processes should be created according to each environment's unique requirements, but when in doubt, the general recommendation is to back up the database every hour and the transaction log every 30 minutes during production time; possibly less often during off-peak.

**Best Practice:** Always archive or mirror snapshot copies.

It is strongly recommended to archive or mirror backup sets as soon as possible. Disasters do occur, and if the storage device containing the databases and backup snapshot images is adversely affected, it will not be possible to restore the databases from the primary storage system. Archive media can be tape, another FAS system, a NearStore device or even local DATA drives available in the IBM System Storage series. The archive process is described in further detail in SMSQL 2.0 installation/administration guide.

**Best Practice: Tips for creating LUNs.**

1. **When specifying a UNC path to a share of a volume to create a LUN, use IP addresses instead of host names.** This is particularly important with iSCSI, as host-to-IP name resolution issues can interfere with the locating and mounting of iSCSI LUNs during the boot process.
2. **Use SnapDrive to create LUNs** for use with Windows to avoid complexity.
3. **Calculate disk space requirements** to accommodate for data growth, snapshot copies, and space reservations.
4. **Leave automatic snapshot scheduling off** as configured by SnapDrive.

**Best Practice: Run Consistency Checker (DBCC CHECKDB) to validate database before and after migration.**

For usual operations, also try to execute the verification process during off-peak hours to prevent excessive load on the production system. If possible, offload the entire verification process to another SQL Server instance running on another server.

**Best Practice: Avoid creating snapshot copies on volumes with active snapshot writable LUNs.**

Creating a snapshot copy of a volume that contains an active snapshot copy is not recommended because this will create a “busy snapshot” issue.

Avoid scheduling SMSQL snapshot copies when

- Database verifications are running
- Archiving a snapshot backed LUN to tape or other media.

**Best Practice: When using supplemental rolling snapshot copies:**

- Be sure to thoroughly read the SMSQL Installation and Configuration guide section “Minimizing your exposure to loss of data”
- Be sure to configure rolling snapshot copies to occur only between SMSQL backup and SnapMirror processes. This will prevent overlapping snapshot schedules.

**Best Practice: Use alternative means for remote management.**

IBM N series recommends avoiding Terminal Server for server management when possible. When using Terminal Server with Windows 2003 (not Windows 2000), you may use a remote desktop to connect if you use a command line parameter/console, shown below:

```
%SystemRoot%\System32\mstsc.exe /console
```

When using Windows 2000, it is advisable to use either NetMeeting remote control or RealVNC (available at [www.realvnc.com/download.html](http://www.realvnc.com/download.html)).

## Appendix B: Consolidation methodologies

Consolidation of SQL Server databases, servers, and storage can provide great return on investment (ROI) benefits and optimize hardware, software, and administrative resources. This section will identify general approaches and methodologies in consolidating SQL Server environments. Much of the information presented in preceding sections will be put into practice in a simulated customer deployment scenario.

### Phase I: Analysis of existing SQL Server instances

This section will highlight some of the key planning phases that should be undertaken in the beginning stages of a SQL Server consolidation project by focusing on a fictitious company, “ACME Corp” which has a total of eleven databases running on four dedicated SQL Servers on direct attached storage (DAS) and wants to consolidate all instances into a two-node cluster configuration with IBM N series storage.

The analysis phase begins with collecting information about existing databases in terms of bottom line criticality to the business. Table 4 illustrates the results of collecting information that is both technical and business related. The High Availability Index is a value used to depict the business-critical characterization of each database and will be used as a deciding factor in determining which backup and SnapMirror policies are implemented for each database. The High Availability Index works on a five-level designation with an assignment of 1 being the most critical and 5 signifying the lowest level of criticality.

SQL Server Instance	Database Name	High Availability Index
SQL Server 1	DB 1	1
	DB 2	1
SQL Server 2	DB 3	2
	DB 4	1
	DB 5	2
SQL Server 3	DB 6	2
	DB 7	1
	DB 8	3
SQL Server 4	DB 9	3
	DB 10	3
	DB 11	1

**Table 4. ACME’s current SQL Server information.**

Table 4 shows a completed table of information describing all databases intended for migration to the new environment along with the ranking of importance to the business. Once the databases have been identified and ranked, the next phase of determining key technical metrics concerned with size, change rate, growth, and throughput can proceed.

It can sometimes be difficult to get data describing I/O throughput during peak loads (holidays, end of quarter, or other peak times), but this data can be very valuable data when sizing storage devices correctly for current and future loads at peak times. Table 5 shows the type of data that can be very helpful when sizing the target storage device.



Database	Database Size	Read per Second (MB)	Write per Second (MB)	Change Rate per Month	Growth Rate per Month
DB 1	270GB	5	1	15%	5%
DB 2	8GB	8	1.5	10%	8%
DB 3	170MB	20	6	12%	6%
DB 4	2GB	0.5	0.1	5%	40%
DB 5	270MB	1.5	0.2	20%	10%
DB 6	43GB	1	0.5	10%	5%
DB 7	500GB	8	2	5%	1%
DB 8	320GB	5	1.2	50%	10%
DB 9	77MB	4	2	5%	5%
DB 10	100MB	2	3	10%	10%
DB 11	3112GB	20	10	10%	5%
<b>Totals</b>	<b>4.872TB</b>	<b>75MB/sec</b>	<b>27.5MB/sec</b>		

**Table 5. ACME's I/O throughputs at peak load.**

Throughput data can be gathered with several native tools provided by the base operating system and the SQL Server database engine. Appendix C uses one such tool by leveraging the function `fn_virtualfilestats` available in both SQL Server 2000 and 2005. The script captures two sets of data points around read/write capacity over a timeframe specified by a DBA. By comparing `the delta bytes written` and `delta bytes read` over say an hour of preferably "peak usage," an approximate number of read and write megabytes per second can be calculated for each database. For detailed analysis, the script can easily be altered to gather multiple data points for a longer period and trended out to see more distinct peaks and valleys.

Growth rate in terms of space can also be calculated based on projected system growth from a user growth capacity and business perspective. Things like database groom tasks, "data churn," snapshot and transaction log retention should also be taken into consideration.

If unsure how to calculate the total number of disks required, please contact IBM N series for an official SQL Server and IBM N series storage sizing request. This will ensure the right number of spindles is designated for your organizations unique workload characteristics.

#### ACME's requirements

- New business guidelines for ACME have dictated that all four SQL Servers be consolidated onto two newly purchased, two-unit servers in an Active/Active cluster with a storage back end of an N series N5500 using eight full shelves of FC disks. A single aggregate model has been selected consisting of a total of 112, 144GB drives using RAID-DP with a group size of 16 disks.

The resulting configuration takes the following things into consideration:

- DB1, DB2, DB7, and DB11 have been designated as requiring the highest level of availability and require a full backup every 30 minutes and a transaction log backup at every 15 minutes with immediate SnapMirror replication to a DR site.
- DB3, DB5, and DB6 do not have as stringent requirements for recovery and can have their transaction logs backed up every 30 minutes with a full backup every hour. Off-site SnapMirror replication is required for DR.

- DB8, DB9, and DB10 have lower BR requirements and can be backed up every hour and will not use SnapMirror.
- DB11, DB5, and DB6 require the best possible configuration for performance.
- All databases will need to keep at least three snapshot copies and additional transaction logs online at all times.
- Management has not budgeted for all databases to be mirrored and have allocated limited space to the DR site.

## Phase II: Designing and migrating to the new environment

Based on the variety of requirements specified by ACME, along with hard data that was collected and displayed in Figure 6 and Table 6, a reference diagram was drawn up for the new SQL Server architecture (see Table 6).

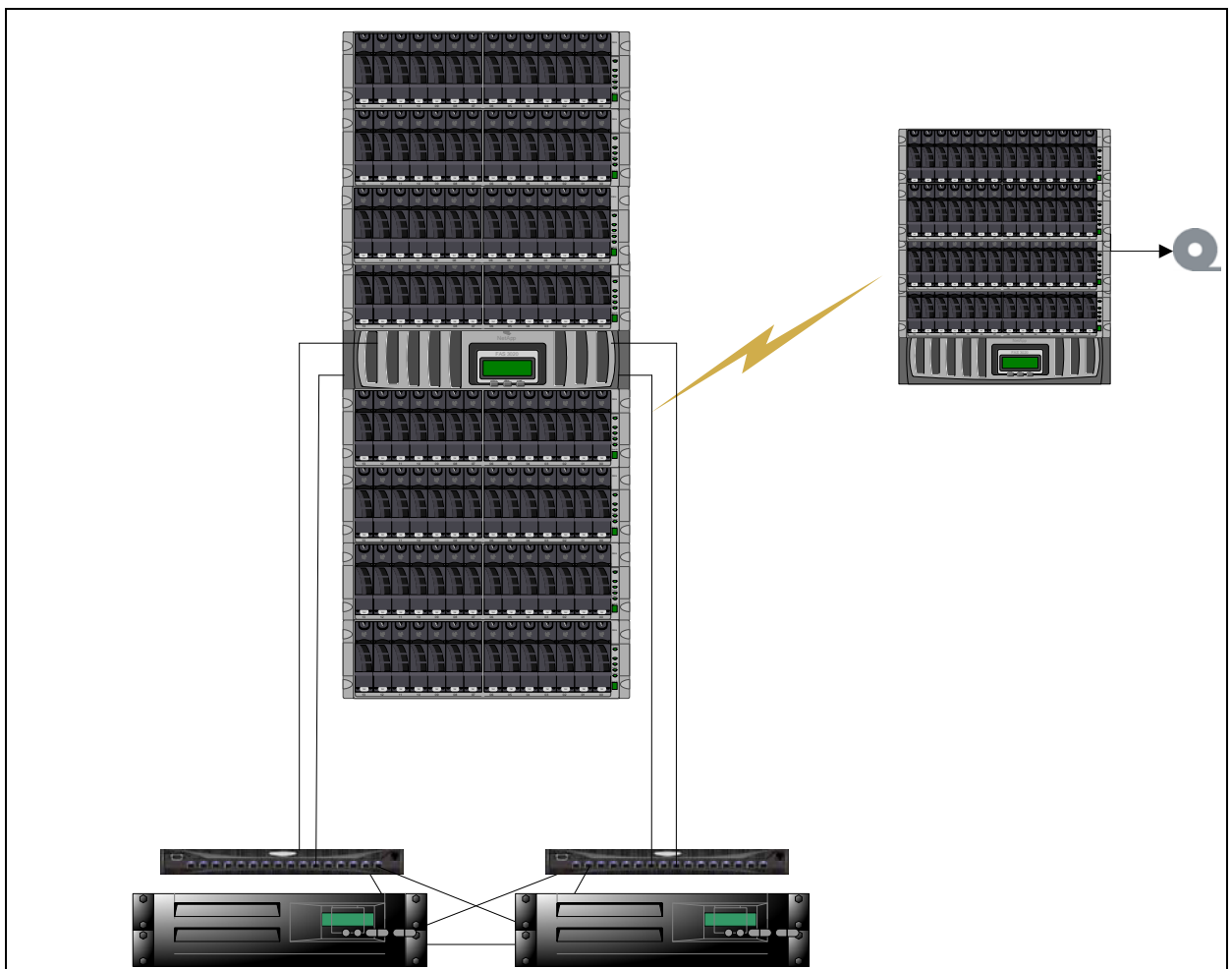


Figure 6. ACME SQL Server environment.



All databases were migrated to the N5500 with rules applied in the following areas:

- **Size and Growth**—All user database LUNs have been sized by a factor of 2X + delta and have been calculated with additional room for growth for the next twelve months. DB11, sized at just over 3TB, has been split into multiple data files and spread evenly over four LUNs.
- **High Availability**—The base aggregate has been constructed using RAID-DP, and dual paths with MPIO have been used for connectivity. MSCS has been implemented for server level failure.
- **BR**—All databases that did not have conflicting requirements from a performance perspective have been logically grouped in FlexVol volumes by BR requirements. When possible, databases were also grouped in FlexVol volumes for granular control over SnapMirror replication.
- **Performance**—Some databases with more stringent performance requirements have had data and log files separated into different FlexVol volumes for optimal performance.

## Consolidation results

Table 6 shows ACME's final consolidation of 11 databases to a two-node Active/Active SQL Server 2005 cluster. The reference diagram shows the hardware configuration that will host the 11 databases. Careful attention will be paid to FlexVol, LUN, and SnapMirror implementation based on different BR, performance, and DR requirements. Table 6 shows the output of all requirements. Then, a recap and notable observations is provided.

Instance	Database	Size	Database Type	H/A Index	LUN Mount (Device)	FlexVol Volume	SnapMirror
Quorum disk	Quorum data	100MB			Q	1	No
Node 1	System DBs	10GB		1	U	2	No
	DB 1	270GB	OLTP	1	E	3	Yes
	DB 2	8GB	OLTP	2	F	4	No
	DB 3	170MB	OLTP	1	G	3	Yes
	DB 4	2GB	OLTP	2	H	5	No
	DB 5	270MB	OLTP	2	I,J	5,6	No
	DB 6	43GB	OLTP	1	K,L	7,8	Yes
	DB 7	500GB	OLTP	3	Y	9	No
	DB 8	320GB	OLTP	3	N	9	No
	DB 9	77MB	OLTP	3	O	9	No
SMSQL	SnapInfo			1	P	10	Yes
Node 2	System DBs	150GB		1	W	11	No
	DB 10	100MB	DSS	2	S	11	Yes
	DB 11	3112GB	DSS	2	M, R, T, V	12,13	Yes
SMSQL	SnapInfo			1	Y	14	Yes

**Table 6. Recommended database placement for ACME migration.**

### Key observations:

- Cluster Server 1 groups databases on FlexVol volumes based on a backup policy of transaction log and database backups every 15 minutes and 30 minutes respectively. Databases are also grouped by FlexVol volume for different SnapMirror policies and frequency.
- DB 5, DB 6, and DB 11 are all using two FlexVol volumes each to separate log and data files for optimal data locality because performance requirements for these databases are more stringent.
- The largest database, DB11, is using two FlexVol volumes for performance with four LUNs. Each data LUN (M, R, and T) has one of three data files located in one file group while the log resides in its own FlexVol volume in LUN V. Note that no other database is using any of these LUNs as dictated by the SMSQL backup rule of sharing beyond two LUNs.
- The system databases for both instances have dedicated FlexVol volumes. The instance on node two is hosting the largest database, DB 11. The system database volume sized at 150GB to allow ample room for growth of `tempdb` for large query sorting and indexing that take place in the DSS database DB 11.
- DB7, DB8, and DB9 are rated 3 on the high availability index and are not configured for SnapMirror. They also all reside on the same volume to prevent them from being backed up with other schedules or unnecessarily mirrored.
- Snapinfo has also been designated for mirroring to ensure that both metadata and transaction log backups are transmitted to the SnapMirror site.
- Because the cluster is Active/Active and must allow for takeover, no drive letter is duplicated on either active node.

The ACME example illustrates a successful implementation using optimal SQL Server file placement, volume, and LUN layout while staying within the confines of drive letter limitations and SMSQL backup requirements. All files, LUNs and aggregates, and FlexVol volumes were implemented to provide just the right level of granular control over performance, backup, and DR policies.



## Appendix C: Sample SQL Server sizing script

---

```
DECLARE @total int
DECLARE @now datetime
select @now = getdate()
select @total = 0

USE master

SELECT @total=count(*)
FROM sysobjects
WHERE (name = 'sqlio')

IF @total = 0 BEGIN
CREATE TABLE SQLIO
  (dbname varchar(128) ,
  fName varchar(2048),
  startTime datetime,
  noReads1 int,
  noWrites1 int,
  BytesRead1 bigint ,
  BytesWritten1 bigint ,
  noReads2 int,
  noWrites2 int,
  BytesRead2 bigint ,
  BytesWritten2 bigint,
  endTime datetime,
  deltawrites bigint,
  deltareads bigint,
  timedelta bigint,
  fileType bit,
  fileSizeBytes bigint
  )
END
ELSE
--remove what was there from before
delete from sqlio

--grab baseline first sample
INSERT INTO SQLIO
SELECT
  cast(DB_Name(a.dbid) as varchar) as Database_name,
  b.filename,
  getdate(),
  cast(numberReads as int),
  cast(numberWrites as int),
  cast(a.BytesRead as bigint),
  cast(a.BytesWritten as bigint),
  0,
  0,
  0,
  0,
```





```
0,
0,
0,
0,
'filetype' = case when groupid > 0 then 1 else 0 end,
cast(b.size as bigint) * 8192
FROM
::fn_virtualfilestats(-1,-1) a ,sysaltfiles b
WHERE
a.dbid = b.dbid and
a.fileid = b.fileid
ORDER BY Database_Name

/*DELAY AREA - change time at will */
WAITFOR DELAY '000:30:00'

UPDATE SQLIO
set
BytesRead2=cast(a.BytesRead as bigint),
BytesWritten2=cast(a.BytesWritten as bigint),
noReads2 =numberReads ,
noWrites2 =numberWrites,
endtime= getdate(),
deltaWrites=(cast(a.BytesRead as bigint)-BytesRead1),
deltaReads= (cast(a.BytesWritten as bigint)-BytesWritten1),
timedelta = (cast(datediff(s,startTime,getdate()) as bigint))

FROM ::fn_virtualfilestats(-1,-1) a ,sysaltfiles b
WHERE fName= b.filename and dbName=DB_Name(a.dbid) and
a.dbid = b.dbid and
a.fileid = b.fileid

/*dump results to screen */
select * from sqlio
```



## Trademarks and special notices

---

© International Business Machines 1994-2007. IBM, the IBM logo, System Storage, and other referenced IBM products and services are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. All rights reserved

References in this document to IBM products or services do not imply that IBM intends to make them available in every country.

Network Appliance, the Network Appliance logo, AutoSupport, Data ONTAP, FlexClone, FlexVol, NearStore, RAID-DP, SnapDrive, SnapManager, SnapMirror, SnapRestore, Snapshot, SyncMirror, WAFL and SnapManager are trademarks or registered trademarks of Network Appliance, Inc., in the U.S. and other countries.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.

Information is provided "AS IS" without warranty of any kind.

All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics may vary by customer.

Information concerning non-IBM products was obtained from a supplier of these products, published announcement material, or other publicly available sources and does not constitute an endorsement of such products by IBM. Sources for non-IBM list prices and performance numbers are taken from publicly available information, including vendor announcements and vendor worldwide homepages. IBM has not tested these products and cannot confirm the accuracy of performance, capability, or any other claims related to non-IBM products. Questions on the capability of non-IBM products should be addressed to the supplier of those products.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.