Symantec Enterprise Vault™

SQL Best Practices

10.0

Last updated: September 6, 2013

Symantec Enterprise Vault™: SQL Best Practices

The software described in this book is furnished under a license agreement and may be used only in accordance with the terms of the agreement.

Last updated: September 6, 2013.

Legal Notice

Symantec Corporation
350 Ellis Street, Mountain View, CA 94043

www.symantec.com

Technical Support

Symantec Technical Support maintains support centers globally. Technical Support's primary role is to help you resolve specific problems with a Symantec product. The Technical Support group also creates content for our online Knowledge Base. The Technical Support group works collaboratively with the other functional areas within Symantec to answer your questions in a timely fashion. For example, the Technical Support group works with Product Engineering and Symantec Security Response to provide alerting services and virus definition updates.

Symantec's support offerings include the following:

- A range of support options that give you the flexibility to select the right amount of service for any size organization
- Telephone and/or Web-based support that provides rapid response and up-to-the-minute information
- Upgrade assurance that delivers software upgrades
- Global support purchased on a regional business hours or 24 hours a day, 7 days a week basis
- Premium service offerings that include Account Management Services

For information about Symantec's support offerings, you can visit our Web site at the following URL:

www.symantec.com/business/support/

All support services will be delivered in accordance with your support agreement and the then-current enterprise technical support policy.

Contacting Technical Support

Customers with a current support agreement may access Technical Support information at the following URL:

www.symantec.com/business/support/

Before contacting Technical Support, make sure you have satisfied the system requirements that are listed in your product documentation. Also, you should be at the computer on which the problem occurred, in case it is necessary to replicate the problem.

When you contact Technical Support, please have the following information available:

- Product release level
- Hardware information
- Available memory, disk space, and NIC information
- Operating system

- Version and patch level
- Network topology
- Router, gateway, and IP address information
- Problem description:
  - Error messages and log files
  - Troubleshooting that was performed before contacting Symantec
  - Recent software configuration changes and network changes

Licensing and registration

If your Symantec product requires registration or a license key, access our technical support Web page at the following URL:

www.symantec.com/business/support/

Customer service

Customer service information is available at the following URL:

www.symantec.com/business/support/

Customer Service is available to assist with non-technical questions, such as the following types of issues:

- Questions regarding product licensing or serialization
- Product registration updates, such as address or name changes
- General product information (features, language availability, local dealers)
- Latest information about product updates and upgrades
- Information about upgrade assurance and support contracts
- Information about the Symantec Buying Programs
- Advice about Symantec's technical support options
- Nontechnical presales questions
- Issues that are related to CD-ROMs or manuals

Support agreement resources

If you want to contact Symantec regarding an existing support agreement, please contact the support agreement administration team for your region as follows:

Asia-Pacific and Japan	customercare_apac@symantec.com
Europe, Middle-East, and Africa	semea@symantec.com
North America and Latin America	supportsolutions@symantec.com

Contents

Chapter 1

Introduction

Sizing and implementing Enterprise Vault requires careful planning to ensure that the product can perform and scale to expectations, and ensure the underlying Enterprise Vault infrastructure is configured to support the required activity.

The Enterprise Vault SQL servers should be sized, tuned and maintained according to Microsoft best practice advice for SQL Server. See the TechNet article “SQL Server best practices”.

This guide discusses some of the SQL Server best practices from an Enterprise Vault perspective and should be used in conjunction with Microsoft advice. This document does not explore any aspects of high availability.

The Enterprise Vault databases contain varied information including configuration information, item metadata, and reporting statistics. In many cases the data forms a key part of feature workflow and this can result in high workloads that will struggle to co-exist with any other database application. During Enterprise Vault implementation you must pay special attention to the SQL servers and their configuration.

This guide assumes that are you are familiar with how to configure and administer Enterprise Vault, SQL Server and associated products. You can obtain more detailed installation and configuration information from the Enterprise Vault documentation. Symantec also publishes many white papers that explore specific Enterprise Vault details.

For advice on Discovery Accelerator sizing and tuning, see the *Best Practices for Implementation Guide*, which is available from the following page:

http://www.symantec.com/docs/TECH159520

Chapter 2

Server requirements

The anticipated loads will help determine the best SQL Server edition and the number and size of SQL Servers that should be deployed. From an Enterprise Vault perspective, the key differences between the Standard and Enterprise or Datacenter editions of SQL Server are the scalability and high availability options.

Enterprise Vault is a feature-rich product and can therefore result in very diverse SQL Server load profiles between customer deployments. As Enterprise Vault scales, additional databases and the associated increase load will require scaling the SQL Servers to meet the load.

It may be necessary to separate out specific Enterprise Vault databases such as the directory and audit databases to dedicated SQL Servers. We also recommend preparing dedicated SQL Servers with reporting services if FSA reporting is to be deployed.

There are many factors which may influence the deployment, but an initial server sizing guideline would be to provision the CPU cores and RAM described in the table below arranged in as many servers as desired, evenly distributing the Enterprise Vault databases if deployed as multiple SQL servers.

	SQL Server requirements
Low load or no end-users (e.g. Journaling only)	Provision 4 cores and 8 GB RAM for every 8 Enterprise Vault servers
Up to 8 Enterprise Vault servers with normal load	Provision 4 cores and 8 GB RAM for every 4 Enterprise Vault servers
More than 8 Enterprise Vault servers with normal load	**Dedicated server for directory and audit database:** Provision 4 cores and 16 GB RAM for every 8 Enterprise Vault servers **Servers for all other databases:** Provision 4 cores and 8 GB RAM for every 4 Enterprise Vault servers

	SQL Server requirements
FSA Reporting	**Dedicated SQL Servers:** Provision 2 cores and 4 GB RAM for every 8 file servers **Note**: The minimum should be 4 cores and 8 GB RAM.

For example, to support 16 Enterprise Vault servers and 64,000 users you might choose to implement:

- 1× Enterprise edition server with 8 processor cores and 32 GB RAM for the Directory database.

Plus one of the following example combinations to host the other databases:

- 4× Standard edition servers each with 4 processor cores and 8 GB RAM.
- 2× Enterprise edition servers each with 8 processor cores and 16 GB RAM.
- 1× Datacenter edition server with 16 processor cores and 32 GB RAM (provided that the network and storage bandwidth can cater for the overall load).

Scaling Enterprise Vault to meet complex requirements, such as wide geographic distribution, may require deploying multiple Enterprise Vault installations (with independent directory and associated databases).

Hardware considerations

We recommend the Enterprise Vault SQL servers run a single SQL Server instance only, with no other applications or services running on the servers. SQL Server needs to fully utilize the server resources, and another application may introduce contention issues that results in undesirable performance.

CPU considerations

The power of a server is not necessarily determined by the CPU speed in terms of cycles per second. Factors such as the server architecture and number and type of processors and cores can provide a far greater benefit over increasing CPU speed.

Hyper-threading technology is claimed to typically provide up to 30% improvement in performance. These processors contain two architectural states on a single processor core, making each physical processor act as two logical processors. However, the two logical processors must share the execution resources of the processor core, so performance gains may not be attained and in some circumstances can even lead to degradation in performance.

Multi-core technology provides similar performance to a comparable multi-CPU server. These processors contain multiple complete processor cores, which act as complete physical processors. Each physical core has its own architectural state and its own execution resources, so the performance gains are reliable.

With the ever-increasing number of processor core combinations and high clock speeds, the traditional x86 Front Side Bus architecture can start to become a bottleneck beyond eight processor cores. A popular and cost-effective method of scaling up the x86 architecture is to use an architecture that supports non-uniform memory access (NUMA). Processors and memory are grouped into nodes that have high-speed local access. However, access to memory co-located with other processor nodes is slower. Therefore, the operating system (and potentially application software) needs to be NUMA-aware and optimized to make the best use of processors, cores, and their associated resources. Windows server and SQL Server support NUMA. See the MSDN article “How SQL Server supports NUMA”.

The recommended number of processor cores can be composed of either physical CPUs or similar combination of multi-core CPUs, but the sizing should not based on hyper-threaded logical cores.

Note: Hyper-threading is not recommended to improve the database performance due to potential performance problems when the database places a load on the memory. See Microsoft Knowledge Base article 322385 http://support.microsoft.com/kb/322385 and the MSDN article "Be aware: To Hyper or not to Hyper" http://blogs.msdn.com/b/slavao/archive/2005/11/12/492119.aspx for further information. If hyper-threading is to be used, particular attention should be paid to the MAXDOP setting as described in KB322385.

In most cases, the SQL Server instance should manage the CPU resources. Do not set the CPU affinity mask unless absolutely necessary, as this can significantly impact the performance. When you run multiple SQL Server instances, the most common reason for setting the CPU affinity mask is to prevent an instance being starved of resources (see “Multiple SQL Server instances” on page 33).

Memory considerations

The recommended memory should be available at each SQL Server instance to ensure the data manipulation does not cause excessive paging to disk both in the Enterprise Vault databases and tempdb, which will quickly degrade the performance.

You can host the databases on either 32-bit or 64-bit (x64 only) platforms. Using an x64-based platform provides more efficient memory utilization and brings many performance benefits.

Install the appropriate edition of Windows Server and SQL Server to support the capacity of memory you have installed. See the Enterprise Vault compatibility charts for supported versions of SQL Server.

Under normal circumstances, you should allow SQL Server to manage the memory dynamically. It may be necessary to change the SQL Server minimum and maximum memory to ensure the memory is used appropriately between SQL Server instances, Reporting services or other co-located services.

If you plan to use a 32-bit SQL server, tune it carefully to make the best use of available memory. The tuning options depend on using the appropriate edition of Windows and SQL Server for the installed capacity of memory.

If a 32-bit SQL server has more than 4 GB of physical RAM, do the following (these settings should not be used on 64-bit servers):

- Enable the operating system Physical Address Extensions boot flag (/PAE).
- Use the following script to enable Address Windowing Extensions (AWE) memory in SQL Server:

```
sp_configure 'show advanced options', 1
RECONFIGURE
GO
sp_configure 'awe enabled', 1
RECONFIGURE
GO
```

Note: This causes SQL Server to reserve all available memory, which has a performance impact on other applications or in a multi-instance SQL Server environment. If this is not desired, set the max server memory option. See SQL Server books on line: Managing AWE Memory.

- If the SQL server has between 4 GB and 16 GB of RAM installed, use the /3GB boot flag. Do not use /3GB with more than 16 GB RAM.

Network considerations

We recommend that the Enterprise Vault SQL servers and Enterprise Vault servers are connected via gigabit network technology. The SQL servers may require multiple network interface cards to support the anticipated loads.

It is also recommended to disable the TCP Chimney Offload, TCP/IP Offload Engine (TOE) or TCP Segmentation Offload (TSO) to prevent network issues. For guidance in disabling these, see Symantec technical article TECH55653.

Storage considerations

It is vital to ensure the storage does not become a bottleneck. By following Microsoft SQL Server best practices, you can ensure that the SQL server is suitably sized. Avoid using network-based storage for the database files.

In most cases, you will need RAID-based storage to achieve your storage requirements. To maintain performance and reliability, consider hardware-based RAID rather than software-based RAID. To achieve redundancy on striped arrays while maintaining performance, consider the RAID scheme carefully.

RAID levels 5 and 6 are popular, cost-effective methods of achieving redundancy while maintaining striped disk read performance. However, writing incurs a cost of four to six physical operations per write. A poorly sized RAID-5 or 6 implementation can significantly reduce the performance of write-intensive activity. Correctly sizing a RAID-5 or 6 implementation to maintain write performance may become more costly than RAID-1+0, and therefore a RAID-1+0 scheme should be considered.

In the case of local or direct attached storage, use multiple controllers supporting multiple channels to distribute the load between the multiple storage locations and provide sufficient throughput. The controllers should also provide a battery-backed read and write cache to aid performance. A minimum of 512 MB controller cache is recommended for local or direct attached storage.

Before you use partitions on a storage area network (SAN), consider the I/O load together with any other applications that are already using the SAN to ensure that the performance can be maintained. Ideally, discuss the implementation with your SAN hardware vendor to ensure that you achieve optimum performance. Typically, you should create LUNs across as many suitable disks as possible, using entire disks rather than partial disks to prevent multiple I/O-intensive applications from using the same disks. When you configure HBAs on the host, ensure that the Queue Depth is set to an optimal value. This should be discussed with the storage vendor.

When you create a basic NTFS volume on a storage device, it is very important to align the volume with the device sector or stripe unit boundaries to prevent unnecessary disk operations that can significantly impact performance. (Dynamic volumes cannot be aligned at time of publication). See the TechNet article “SQL Server best practices” for more information and using the diskpart tool to create and align volumes. This article also recommends that you format both log and data partitions with 64 KB allocation unit sizes.

In most cases you should only create a single volume on each disk array to avoid contention at the disks between the partitions.

Each database requires the disks to be arranged for two different purposes; the database data files and the transaction log files. The data files require good random access, and therefore a striped array of many disks should be used. The log files

require good sequential write performance, so each log file should be placed on its own high speed array with good transfer rates.

To achieve redundancy on the sequential write-intensive disks (log), use a RAID-1 or RAID-1+0 scheme with high speed, 15k rpm disks.

Arrange the SQL server storage to accommodate the different types of data, distributing the load as appropriate. The following arrangements of storage might be considered for each data requirement:

Recommended partitions for SQL servers

Partition	RAID array
System drive	RAID-1 array
Tempdb log file	RAID-1 or 1+0 array
Tempdb data files	RAID-1+0 array
Directory Database log file	RAID-1 or 1+0 array
Directory Database data file	RAID-1+0 array
Each vault store database log file	RAID-1+0 array
Each vault store database data file	RAID-1+0 array
Each fingerprint database log file	RAID-1+0 array
Each fingerprint database data files	1 or more RAID-1+0 arrays to host the 32 filegroup data files
Audit database log file	RAID-1+0 array
Audit database data file	RAID-1+0 array
Monitoring database log file	RAID-1, 5, or 1+0 array
Monitoring database data file	RAID-5, or 1+0 array
FSA Reporting database log files	RAID-1+0 array
FSA Reporting database data files	RAID-1+0 array
SQL Server Reporting log file	RAID-1, 5, or 1+0 array
SQL Server Reporting data file	RAID-5, or 1+0 array
SQL Server Reporting TempDB log file	RAID-1, 5, or 1+0 array
SQL Server Reporting TempDB data file	RAID-5, or 1+0 array

If multiple database files are located on one partition, it may require regular file defragmentation to maintain performance.

Virtualized infrastructure

There are important aspects to consider when installing SQL Server in a virtualized infrastructure. Follow the recommendations of your hypervisor vendor and Microsoft when you size and configure the environment.

The primary objective is to ensure that the resource requirements described above are dedicated to the virtual machine to ensure minimum impact to the performance from intermediate layers or co-existing guests.

The hypervisor should be type-1 (native) to ensure the minimum impact on hardware resource requirements.

Note the following general guidelines:

- In a typical virtualized infrastructure, local disks might be used for the hypervisor and SAN-based storage for the guest operating system images and data file locations. The operating system and data storage partitions should be independent dedicated locations, as described above.

  Disk partitions should be aligned with the device sector or stripe unit boundaries to prevent unnecessary disk operations that can significantly impact performance.

  The disk partitions to be used for the database log files should be created as recommended by the hypervisor vendor for sequential access (possibly raw hard disks).

  The disk partitions to be used for the database data files should be created as recommended by the hypervisor vendor for random access (most likely virtual hard disks).
- Virtual hard disks should be created as fixed size and not dynamic.
- Avoid the use of hyper-threading by the hyper-visor.
- Avoid the use of virtual machine snapshots, which can impact performance.
- The memory requirements recommended above should be dedicated and prioritized to the virtual machine to prevent dynamic allocation or sharing.
- The number of processor cores as recommended above should be exclusively dedicated to the virtual machine, and the processor priority and bandwidth set to provide the virtual machine with full utilization of the selected CPUs.

If you want to install the SQL Server instance on a virtualized machine, avoid installing multiple instances on the same virtual machine.

Anti-virus considerations

The use of anti-virus products may be necessary to protect company assets, however without tuning some anti-virus products can be very invasive and considerably impact performance. It is vital to ensure any anti-virus product in use is tuned accordingly and key disk locations excluded from real-time scanning.

See the following article for more information:

http://www.symantec.com/docs/TECH176828

Chapter 3

Database sizing

The storage capacity of the Enterprise Vault databases needs to be carefully considered. There are many factors which will influence the sizes of the databases and in some cases it is not practical, because the information is not readily available, to incorporate all of those factors to provide an accurate sizing. The following sections describe calculating a high-level capacity estimate that provides guidance but in practice the actual sizes may vary.

Directory database

The Enterprise Vault directory database stores details of the following:

- Enterprise Vault server configuration
- Enterprise Vault services configuration
- Enterprise Vault policies
- Content source server details and their configuration
- Target object and provisioning or synchronization configuration
- Archive and folder structure details
- Security identifiers and access control lists
- Index location and index volume configuration
- Storage sharing group, vault store and partition configuration
- Archive, PST and Index management tasks

Enterprise Vault 10.0.4 onwards also stores:

- Archive legal hold details
- Enterprise Vault Extension content source and provider registrations
- Enterprise Vault Extension custom archive type registrations

The Directory database is used by all Enterprise Vault servers to co-ordinate all activities, and therefore can be under considerable load.

Use the following rule of thumb to size the Directory database:

Estimated capacity (MB) = ((2.496s)+(23.056u)+((fa+fd)1.3)+(1.613fa)+(0.156fdp))/1000

Where:

s	Total Enterprise Vault servers
u	Total mailboxes
f	Total file servers or SharePoint servers
a	Average archive points per file server or sites and sub-sites per SharePoint server
d	Average folders per file server or libraries/folders in all sites per SharePoint server
p	Average permissions per folder

Note: This calculation provides a high-level estimate only that will vary by feature usage. It does not include growth from archive or folder changes and management tasks. In addition, it does not take into account the operating system, paging, and log file devices, and it does not include any additional capacity that may be required during product version upgrades.

The IO load will depend upon the total archiving and client loads across all Enterprise Vault servers and the SQL Server specification; however the Directory data file on a 16 GB SQL Server under typical loads from 5 Enterprise Vault servers might average 20 IOPS and the log file average 100 IOPS.

The following typical example shows the estimated directory database size for a mailbox archiving environment with 5 Enterprise Vault servers managing a total of 20,000 mailboxes.

Estimated size (MB) = ((2.496s)+(23.056u)+((fa+fd)1.3)+(1.613fa)+(0.156fdp))/1000
= (2.496*5)
+ (23.056*20000)
+ ((0*0)+(0*0)*1.3)
+ (1.613*0*0)
+ (0.156*0*0*0)
/1000
=461.14 MB

Therefore, the estimated capacity of the Directory database needs to be 461.1 MB, plus 20% extra headroom. The total is 553.3 MB for the data file.

In the following large scale example there are to be 20 Enterprise Vault servers which will be archiving a combination of Exchange mailboxes and File Servers. The Exchange archiving will be managing a total of 16,000 mailboxes. The file server archiving will be managing 80 files servers, each containing on average 350,000 folders that typically have 30 permissions on each folder. On average 850 archive points will be created on each file server.

Estimated size (MB) = ((2.496s)+(23.056u)+((fa+fd)1.3)+(1.613fa)+(0.156fdp))/1000

= (2.496*20)

+ (23.056*16000)

+ ((80*850)+(80*350000)*1.3)

+ (1.613*80*850)

+ (0.156*80*350000*30)

/1000

=168,007.03 MB

Therefore, the estimated capacity of the Directory database needs to be 168GB, plus 20% extra headroom. The total is 201GB for the data file.

Audit database

The Enterprise Vault audit database stores details of the following:

- Admin and synchronization configuration changes
- Archive activity (archive item, view item, delete item, restore item) [detailed]
- Item changes to folder location or retention category
- Item recovery
- SharePoint services activity including item retrieval
- Migration activity (PST/NSF) [detailed]
- Advanced searches
- Move archive operations and tasks
- Indexing task operations

All categories provide summary information and several categories can also provide more detailed information. Enabling per item auditing can cause the audit database to grow rapidly, and may also impact performance.

This database is created on the same SQL server as the Enterprise Vault directory database. However, the audit database can be moved to another server if required. If the database should be moved, the EVAudit ODBC system Data Source Name should be updated on each Enterprise Vault server. See the following article for guidelines on how to do this:

http://www.symantec.com/docs/TECH35744

Use the following rule of thumb to size the Audit database. The base annual capacity should be included plus any additional capacity required for the selected audit levels:

Base annual capacity (MB) = (0.59r)+(2.359u)
+Admin audit annual capacity (MB) = 88.1+(176.118u)+(176.118f)
+Archive item annual capacity (MB) = (0.339mt)+(88.1a)
+View item annual capacity (MB) = (88.06a)+(0.339mt)+(22.1u)
+Delete item annual capacity (MB) = (88.1d)
+Tasks annual capacity (MB) = (1.7m)+(0.678i)
+Migration annual capacity (MB) = (88.1p)
/1000

Where:

u	Total users
r	Total archives
f	Average folders added/moved per year across all files servers or SharePoint
a	Average number of items archived per day
d	Average number of items expired/deleted per day
p	Average number of items imported via PST/NSF migration per day
i	Average number of indexes upgraded or rebuilt per year
m	Average number of archives moved per year
t	Average number of items per archive re-indexed or moved

Note: This calculation provides a high-level estimate only. It does not take into account the operating system, paging, and log file devices. It also does not include any additional capacity that may be required during product version upgrades.

The IO load will depend upon the level of auditing, the total archiving and client loads across all Enterprise Vault servers and the SQL Server specification.

Monitoring database

The Enterprise Vault monitoring database stores details of the following:

- Monitoring counter, collection and threshold configuration
- Gathered metric values and historic records

The monitoring database will retain between 30 and 60 days of metrics sampled at a frequency of between 10 minutes to 1 hour from all monitored Enterprise Vault servers and journal archive sources.

Once the monitoring database has reached its configured maximum size it should then remain reasonably static.

Use the following rules of thumb to size the Monitoring database:

Estimate (MB) = ((46.88j)+(15.63e)+(9.38v)+(51.05s)+(r(1440/f)*((9.03j)+(1.88s)+(0.952e))))/1000

Where:

s	Total enterprise vault servers
e	Total Exchange servers
j	Total journal mailboxes
v	Total Vault Stores
r	Retention period (30, 45 or 60 days – default is 30 days)
f	Sample frequency (10 to 60 minutes – default is 15 minutes)

Note: This calculation provides a high-level estimate only. It does not take into account the operating system, paging, and log file devices. It also does not include any additional capacity that may be required during product version upgrades.

The IO load at the Monitoring database is likely to be low with regular peaks during statistic updates.

The following example describes an environment with 5 Enterprise Vault servers and a total of 10 vault stores, archiving 10 Exchange servers with 10 journal mailboxes at the default sample rate of every 15 minutes and retained for the default duration of 30 days.

Estimate (MB) = ((46.88j)+(15.63e)+(9.38v)+(51.05s)+(r(1440/f)*((9.03j)+(1.88s)+(0.952e))))/1000

= ((46.88*10)

+ (15.63*10)

+ (9.38*10)

+ (51.05*5)

```
 + ( (30*(1440/15)
 * ( (9.03*10)
   + (1.88*5)
   + (0.952*10) ) )
 )/1000
=315.53 MB
```

Therefore, the estimated capacity of the Monitoring database needs to be 315.53 MB, plus 20% extra headroom. The total is 378.6 MB for the data file.

Enterprise Vault Reporting database

Enterprise Vault reporting uses SQL Server Reporting Services to provide the reports. The reports, which can be customized, contain various details including:

- Enterprise Vault service and task status
- Volume of items archived per Enterprise Vault server
- Mailbox archiving status
- Archive quota usage per user
- Most frequently accessed archived items (requires audit database)
- Journal mailbox archiving status and trends (requires monitoring database)
- Vault store usage by archive or billing account

Some of the report content is generated from the Enterprise Vault monitoring or audit databases. See the Enterprise Vault documentation for more details of which reports require either the monitoring or audit databases.

The SQL Server Reporting Services and databases should be sized according to Microsoft recommendations.

Vault Store database

The Vault Store database stores details of the following:

- All items archived within the vault store and associated metadata
- Archive statistics and, in Enterprise Vault 10.0.4 and later, content provider statistics
- Items requiring indexing, backup or synchronization of client applications
- Items requiring updates such as folder path or retention category
- Items requiring deletion or expiry
- Items currently in processing by the indexing services

- Item legal hold status details

The maximum number of items that may be stored in a single Vault Store database during its lifetime is 2,147,483,647. This includes items that have expired and are no longer in the database. Most customers are unlikely to reach this limit but a customer archiving a million items a day to a single Vault Store could reach this limit after 5 to 6 years of archiving irrespective of expiry policy. You may check how many items have been stored by running this SQL statement against the vault store database:

```
SELECT IDENT_CURRENT(N'Saveset') AS NextSavesetIdentity
```

In addition, a single archive cannot perform more than 2,147,483,647 data change operations such as archive, update and expire in a single Vault Store. This is most likely to occur with journal archives and the best method of managing this situation would be to roll over journal archives on a regular basis (either by archived volume or date). You may check to find the most operations that have been performed for any archive in a Vault Store by running the following statement against the vault store database:

```
SELECT MAX(HighestIndexSeqNo) From ArchivePoint
```

In general, we recommend that a vault store does not contain more than 250 million items. This makes it easier to perform database maintenance activities such as rebuilding indexes and backing up databases.

Use the following rules of thumb to size each Vault Store database:

Base capacity (MB) = ((0.12m*0.5)/1024)+4096

Yearly item metadata (MB) = (0.5*(m+ j) + 0.7d + 3f)/1024

Yearly device metadata (MB) = ((m+j+d+f)*v*0.05)/1024

Where:

m	Total Exchange mailbox messages archived into vault store per year	
j	Total Exchange journal messages archived into vault store per year	
d	Total Domino messages archived into vault stores per year	
f	Total file based items archived into vault store per year	
v	Vault store storage device type, where:	
	Local/CIFS/SMB	= 0
	Streamer	= 1
	EMC Centera (without collections)	= 10

Note: This calculation provides a high-level estimate only. It does not take into account the operating system, paging, and log file devices. It also does not include any additional capacity that may be required during product version upgrades.

The IO load will depend upon the archiving and client loads and the SQL Server specification; however each Vault Store data file on a 16 GB SQL Server under typical loads might require 150 – 1,200 IOPS perhaps averaging 300 IOPS, whilst the log file IO load might average 100 IOPS.

Fingerprint database

The Vault Store database stores details of the following:

- All sharable item content hash values
- All sharable item shared link references

Typically, all files or mail attachments larger than 20 KB will be identified as sharable and a document identity hash value added to the fingerprint database. For message based archiving this normally represents around 20% of the messages. File based archiving of office type documents are likely to see a similar proportion, but non-office files may vary considerably depending upon the data source.

Use the following rule of thumb to size the Fingerprint database. It assumes a message sharing ratio of 2 and file sharing ratio of 1.2.

Capacity per year (MB) = (((pm/2)+(f/1.2))*0.5)+((m+f)*v*0.05) /1024

Where:

m	Total messages archived per year in all vault stores in sharing group	
p	Percentage of messages with attachments eligible for sharing (assume 0.2 if unknown)	
f	Total files archived per year in all vault stores in sharing group	
v	Vault store storage device type, where:	
	Local/CIFS/SMB/Centera	= 0
	Streamer	= 1

Note: This calculation provides a high-level estimate only. It does not take into account the operating system, paging, and log file devices. It also does not include any additional capacity that may be required during product version upgrades.

The IO load will depend upon the total archiving and client loads across all Vault Stores in the sharing group and the SQL Server specification; however the Fingerprint data files on a 16 GB SQL Server under typical loads from 5 Enterprise Vault servers might average 80 IOPS and the log file average 120 IOPS.

As the Enterprise Vault requirements scale it may be necessary to distribute the I/O load of the fingerprint database. The Fingerprint database partitions its tables across 32 file groups and these can be distributed across multiple disk partitions

during creation using the “New Vault Store Group” wizard. If changes to the locations are required after deployment, the filegroups would need to be moved manually within SQL Server.

FSA Reporting database

The FSA Reporting database stores details of the following:

- Server and drive statistics
- Folder and file statistics
- User statistics

The FSA Reporting databases enable the data analysis reports to be generated and are updated by the Enterprise Vault File Collector service. The SQL Server Reporting Services and databases should be sized according to Microsoft recommendations.

If required, at least one FSA Reporting database should be installed per directory database. Multiple FSA Reporting databases can be created on separate SQL Servers to scale-out the anticipated loads. Multiple FSA Reporting databases can also be used to segregate data, for example by geographical region.

Ensure that the file servers are evenly distributed between the FSA Reporting databases. The amount of data that the File Collector server must upload from a scan of a file server increases with the number of archive points and volumes that require scanning.

A data upload bottleneck can result if you assign many file servers that have the same FSA Reporting scan schedule to the same FSA Reporting database. If possible, distribute file servers with the same scan schedule to different FSA Reporting databases. Alternatively stagger the FSA Reporting scan schedules for the file servers that are assigned to the same FSA Reporting database.

Note: The FSA Reporting databases require special maintenance attention to ensure they remain within acceptable sizes.

FSA Reporting creates a SQL Agent job to move data from current tables to historical tables, by default at 9PM each day, but if the SQL Agent or purge job is disabled the current tables could grow unconstrained.

The historical tables will also require regular manual trimming to prevent unlimited growth. See the Enterprise Vault documentation for more information on using the FSAReporting_TrimData.bat utility to trim the historical tables. It is recommended to keep all historical tables to within 30 days data.

Use the following rules of thumb to size the FSA Reporting database:

Estimated (MB) = ((4.248f)+(0.339fv)+(2.218fr)+(0.536tr)+(0.61fvur)+(0.11fvr))/1000

Where:

s	Total number of file servers
u	Average number of users per file server
v	Average volumes per file server
t	Total number of unique file types
r	Historical retention period (days – recommended 30)

Note: This calculation provides a high-level estimate only. It does not take into account the operating system, paging, and log file devices. It also does not include any additional capacity that may be required during product version upgrades.

The IO load will depend upon the file collector loads and the SQL Server specification; however each Vault Store data file on an 8 GB SQL Server under typical loads might require 150 – 1,200 IOPS perhaps averaging 300 IOPS, whilst the log file IO load might average 100 IOPS.

Chapter 4

Deployment and tuning

The SQL server and databases require additional tuning to ensure the best performance is achieved.

Deployment

SQL permissions

The Enterprise Vault administration features enable Enterprise Vault administrators to create various databases. This is dependent on the Vault Service account having the SQL server role of database creator (dbcreator).

The Enterprise Vault databases are created with the Vault Service account as the DBO which provides sufficient permissions for normal operation. See the following article for the minimum permission requirements:

http://www.symantec.com/docs/TECH65841

Enterprise Vault 10.0.3, and Compliance Accelerator and Discovery Accelerator 10.0.3 introduce database roles which you can use to improve SQL database security.

You can use these roles to grant the Vault Service account only the permissions needed for normal daily operations, and additional permissions when they are required. See the following article for more information:

http://www.symantec.com/docs/HOWTO80670

Deploying databases and the impact of model database

SQL Server creates new databases based upon the model database, so any changes to the model database will also be present in the Enterprise Vault databases. In addition, some of the default model database values may not be appropriate for some of the Enterprise Vault databases, for example, options such as file autogrowth values.

Therefore after creating any Enterprise Vault databases, some of the options will need to be checked. The following settings and options can be examined and changed by either opening the database properties in SQL Management Studio or through the use of database views, sp_dboption and ALTER DATABASE (see SQL Server Books Online).

- The database transaction log file autogrowth should be set around 200 MB – 1 GB for each Enterprise Vault database. The database data file autogrowth value should be set according to the recommended values in the table below. This can be viewed using SQL Management Studio database properties or the following SQL statement could be used to gather the sizes (in 8 KB pages):

  ```
  SELECT name, type_desc, physical_name, size, growth, max_size,
  is_percent_growth from sys.database_files
  ```

  Recommended auto growth values for databases

Database	Data file autogrowth value
Directory	Approximately 1 – 2 weeks growth as per directory sizing for synchronization. For example, 200 MB growth, unlimited growth.
Vault Store	Approximately 1 – 2 weeks growth as per sizing for daily archiving. For example, 500 MB growth, unlimited growth.
Fingerprint	50 MB growth per filegroup, unlimited growth
Monitoring	200 MB growth, unlimited growth
Audit	300 MB growth, unlimited growth
FSA Reporting	200 MB growth, unlimited growth

- The database recovery model should be set according to your backup or high availability strategy. The recommended default is FULL. The current value can be viewed using SQL Management Studio database properties or using the following SQL statement:

  ```
  SELECT name,(CASE recovery_model WHEN 1 THEN 'FULL' WHEN 2 THEN
  'BULK_LOGGED' WHEN 3 THEN 'SIMPLE' END) from sys.databases
  ```

- The database options should be checked with the expected default values. The current options can be viewed using SQL Management Studio database properties dialog or the following SQL statement will show the options set:

  ```
  EXEC sp_dboption <database name>
  ```

  Only the AUTO_CREATE_STATISTICS and AUTO_UPDATE_STATISTICS options should be set. Any other options set should be returned to their default using either the SQL Server Management Studio database properties dialog, or using ALTER DATABASE (see SQL Books Online).

When Enterprise Vault creates each database they will be created with the following default sizes. It is recommended to increase the data device size to either the first year's growth or at least the expected initial growth. This prevents file fragmentation, wasted I/O and waits growing the files during the initial, potentially high loads, archiving any backlog.

Partition	Transaction Log	Data device
Directory	25 MB	10 MB
Vault Store	80 MB	100 MB
Fingerprint	80 MB	Primary filegroup 100 MB Non-primary filegroups 32 × 1 MB
Monitoring	80 MB	100 MB
Audit	80 MB	100 MB
FSA Reporting	80 MB	100 MB

Co-locating Enterprise Vault databases

The Enterprise Vault databases can be co-located on the same SQL server, provided the appropriate resources are available; however hosting many very active databases on a single SQL server can become detrimental to performance.

When co-locating multiple databases on a single SQL Server it is important to ensure the expected I/O loads can be accommodated by the attached storage. This may require distributing the database files between multiple storage arrays to isolate the loads.

You must size the SQL server appropriately to host the required load. The cost of scaling up a single server can become less effective than using multiple SQL servers.

The Enterprise Vault databases should be distributed between the number of SQL servers identified in the sizing exercise. The sizing exercise may have identified the need for a dedicated server for the Directory database and FSA Reporting databases. If per-item auditing is required and there are more than 5 Enterprise Vault servers it would be worth considering hosting the audit database on a dedicated SQL server.

It is recommended not to co-locate non-Enterprise Vault databases on the same SQL server.

Database tuning

The tempdb database

The tempdb database is a shared resource that is used to store the following:

- User objects (user defined objects, temporary tables and indexes, table variables, tables returned in functions, and system tables).
- Internal objects (work tables for cursors, spool operations, LOB storage, hash join storage, hash aggregate storage, and intermediate sort results).
- Version stores (row versions from updates).

The tempdb database is a temporary database containing transient data that is recreated on each restart of SQL Server, so it will not need to be recovered.

The tempdb database pages move quickly between memory and disk, which becomes a bottleneck if the disks are inappropriate and the configuration is not tuned. You can take the following steps to avoid problems:

- To prevent tempdb file growth causing unnecessary I/O, and to ensure that the tempdb files do not become severely fragmented, set the minimum file sizes to at least 200MB and set them to auto-grow by 10%.
- Move the tempdb data files to a dedicated striped disk array (RAID-1 or 10) and the log file to a dedicated high speed drive with high transfer rates (RAID-1 or RAID-10). You can use the following SQL statement to move the file location, which will take effect on the next SQL Server restart:

```
USE master;
GO
ALTER DATABASE tempdb
MODIFY FILE (NAME = tempdev, FILENAME = 'E:\SQLData\tempdb.mdf');
GO
ALTER DATABASE tempdb
MODIFY FILE (NAME = templog, FILENAME = 'F:\SQLData\templog.ldf');
GO
```

Warning: The tempdb database is recreated each time SQL Server is started. If the new filenames are invalid, SQL Server will fail to start. When this happens, you must start SQL Server from the console with the –f flag to enter single user mode. Then you can repeat the above procedure with a valid location.

- Create one data file per CPU core, taking into account CPU core affinity mask settings (to spread the load and reduce contention), making each file the same size as above.
- Do not replicate the tempdb storage device or configure SQL Server replication on the tempdb database. The tempdb database is a temporary database containing transient data that is recreated on each restart of SQL Server, so it will not need to be recovered. Any form of replication can significantly impact the performance.

Multiple SQL Server instances

Running multiple SQL Server instances on the same server is not recommended because of the load profile of Enterprise Vault. However, if this is unavoidable, you should tune the instances appropriately to reduce any impact between them and ensure that they are not starved of resources.

The SQL Server instance that hosts the Enterprise Vault databases should have the minimum memory option configured to ensure that the memory recommended in the "Memory considerations" section on page 13 is dedicated to it.

To ensure that the SQL Server instance is not starved of CPU resource, you may need to consider balancing the available CPUs between the installed SQL Server instances. There are several ways to do this, including the following:

- Using the Windows 2003 or Windows 2008 resource manager to allocate resources to each instance.
- Setting the CPU affinity mask of each SQL Server instance to bind each SQL Server scheduler to specific CPUs.

You can also limit a SQL Server instance to a subset of CPUs without binding the SQL Server scheduler to specific CPUs by using trace flag 8002 with CPU affinity. This should counteract any performance problems that are associated with using CPU affinity. For more information, see the Microsoft Press book on SQL Server tuning.

The hardware architecture needs to be considered to ensure CPUs are not inefficiently allocated to the SQL Server instances, which can be significantly detrimental to performance.

In the case of NUMA architectures, the processors, cores, and memory are arranged in nodes that should be carefully allocated to the SQL Server instances to avoid

creating a memory or I/O bottleneck. It may be worth considering using Soft-NUMA to divide the resources with both non-NUMA and NUMA-based hardware. See the Microsoft Press books for more information.

Reporting services

Reporting services may be required for Enterprise Vault reporting or FSA Reporting. If the reporting services are not located on a dedicated SQL Server, it may be necessary to tune the memory requirements for both the SQL Server engine and Reporting services to ensure the available resources are appropriately shared.

For more information on configuring reporting services memory limits, see the MSDN article “Configuring Available Memory for Reporting Services”.

Advanced tuning

There are circumstances in which the SQL Server may benefit from additional tuning, taking more advanced approaches which may or may not be beneficial.

Distributing I/O with multiple database files

Adding multiple database files to Enterprise Vault databases should not be necessary and the recommended approach is to ensure the Enterprise Vault database files are placed on an adequately sized storage array.

Moving tables or indexes into file groups

Splitting tables and indexes into separate file groups can add considerable complexity to the database management. Any errors during implementation can lead to data loss, damage to the schema, and problems during product upgrades. Therefore, making changes to the existing schema is not recommended.

However the Fingerprint database does partition tables across 32 file groups and these can be distributed across multiple disk partitions using the “New Vault Store Group” wizard. If changes to the locations are required after deployment, the filegroups would need to be moved manually within SQL Server.

Chapter 5

Maintenance

Implementing an Enterprise Vault database maintenance plan is essential to preventing data loss, protecting data integrity and preserving the databases performance.

Maintenance plan

It is recommended regular backups of all Enterprise Vault databases are performed. When performing backups the Enterprise Vault services should be placed into read-only/backup-mode, and all other associated data backed up to ensure consistency across the backups.

After the backup has completed the database log files should be shrunk.

As the databases are used, crucial tables and their indexes fill and can become fragmented. This reduces overall performance and can lead to additional I/O. The solution to fragmented indexes is to reorganize or rebuild very active indexes regularly. Index maintenance should be executed whilst Enterprise Vault servers are in read-only/backup-mode.

The following SQL statement can be used to identify all tables and indexes where the external fragmentation exceeds 5% and the table consists of at least 1,000 pages.

```
SELECT OBJECT_NAME(i.object_id) AS TableName,
i.name AS TableIndexName, phystat.avg_fragmentation_in_percent
FROM sys.dm_db_index_physical_stats(DB_ID(), NULL, NULL, NULL,
'DETAILED') phystat
JOIN sys.indexes i ON i.object_id = phystat.object_id AND i.index_id =
phystat.index_id
WHERE phystat.avg_fragmentation_in_percent > 5 AND phystat.page_count >
1000
```

Rebuilding indexes can be time-consuming, so low levels of fragmentation should be addressed with a reorganize instead. The following table recommends the most appropriate action to take according to the level of fragmentation:

Fragmentation	Corrective approach
5% – 30%	Reorganize
More than 30%	Rebuild (Can be achieved as online rebuild)

In addition, as some tables grow very large the statistics update frequency may become less than desirable, resulting in out-of-date statistics (or if auto update statistics has been disabled), potentially causing inefficient query plans. Therefore it is recommended to update the statistics on a regular basis.

The system stored procedure sp_updatestats (available in Maintenance Plans) should be executed on a regular basis to ensure all statistics are up to date. However in some circumstances this may not update all desired statistics. The best way to perform these optimizations is through monitoring to identify any statistics which may require updating.

A method of identifying whether statistics should be manually updated using UPDATE STATISTICS might be to execute the following query:

```
SELECT object_name(i.object_id) as 'table_name',
       i.name as 'index_name',
       STATS_DATE(i.object_id, i.index_id) as 'last_stat_update',
       (case WHEN rowcnt>rowmodctr THEN abs((1-(cast(rowmodctr AS
float)/cast(rowcnt AS float)))*100) WHEN rowcnt<rowmodctr THEN
abs((1-(cast(rowcnt AS float)/cast(rowmodctr AS float)))*100)
ELSE 0.0 END) as 'percent_change'
FROM sys.indexes i
LEFT JOIN sys.objects o ON o.object_id = i.object_id
LEFT JOIN sys.sysindexes si ON o.object_id = si.id
WHERE o.type='U' AND rowmodctr>0 AND rowmodctr<rowcnt
AND (case WHEN rowcnt>rowmodctr THEN abs((1-(cast(rowmodctr AS
float)/cast(rowcnt AS float)))*100) WHEN rowcnt<rowmodctr THEN
abs((1-(cast(rowcnt AS float)/cast(rowmodctr AS float)))*100)
ELSE 0.0 END) > 5
```

Note: As of SQL 2005 the sysindexes view is provided for compatibility purposes, and the rowmodctr value is a calculated value which may not be accurate. However it should still be sufficient to indicate outdated statistics.

For example, the following tables (and in particular their indexes or statistics) benefit from specific maintenance. These tables are very active and have a number of indexes to maintain.

Database	Table	Notes
Directory	ACE	Used to store security descriptor references. Likely to have a higher read-to-write ratio.
Vault Store	Saveset	Used to store all archived items. Likely to have an equal or higher write-to-read ratio.
Vault Store	JournalArchive	Used during archiving process and as part of vault cache workflow. This table will have a very high turnover but will remain quite large. Likely to have an equal or higher write-to-read ratio.

The database file placement on disk can also lead to file fragmentation, which can degrade performance. If multiple database files reside on a single partition and tend to grow regularly, the database data and log file disks may need to be regularly file defragmented to maintain performance.

Note: The database maintenance plan should not include a data file shrink, to avoid unnecessary file growths. However the database log files may need to be shrunk after backing up.

Moving databases

It may be necessary to move the Enterprise Vault databases, for example to scale out to additional SQL servers, or to retire a SQL server. See the following article for guidelines on how to do this:

http://www.symantec.com/docs/TECH35744

Database upgrades

The version of SQL Server running Enterprise Vault databases may need to be upgraded. It is recommended the following steps are included in the upgrade:

- Backup all Enterprise Vault databases
- Perform the SQL Server upgrade
- Check the Enterprise Vault database compatibility levels (at least level 90)
- Rebuild indexes
- Update statistics with full scan

Enterprise Vault product upgrades will most likely require upgrading the database schemas. The upgrade process may require significant additional storage capacity during the upgrade, most notably at the transaction logs and tempdb database. It is recommended the following steps are included in the upgrade:

- Backup all Enterprise Vault databases
- Temporarily increase the tempdb database storage or enable auto-growth
- Change the Enterprise Vault databases recovery models to “SIMPLE”
- Perform the Enterprise Vault database upgrades
- Rebuild indexes
- Update statistics
- Return the recovery model to its original setting – recommended “FULL”
- Backup all Enterprise Vault databases

Rolling over databases

As the Enterprise Vault environment grows, it may become necessary to roll over certain databases to remain within the available storage capacity and to maintain performance.

Audit database

The auditing database can grow significantly in size depending upon the audit collection settings and the Enterprise Vault loads. The audit database storage requirements can be managed by either removing unwanted data, or rolling over to a new database. Before making any changes it is recommended to make a backup of the EnterpriseVaultAudit database.

The AuditTrail table will grow significantly and can be managed by deleting unwanted rows. However as this table can become extremely large, running a query to delete rows based upon their date will be very expensive and time consuming.

Therefore the best approach would be to backup and then truncate the existing table. See Symantec technical article TECH35746, “How to perform a rollover of the Enterprise Vault auditing database”.

The audit database can be rolled over to a new database, which requires updating the ODBC Data Source Name on each Enterprise Vault server. See Symantec technical article TECH35746, “How to perform a rollover of the Enterprise Vault auditing database”.

Vault Store database

The number of vault store databases required should be determined during the Enterprise Vault solution design to meet the desired environment scalability. However over time it may become necessary to roll over vault store databases to maintain databases within the recommended size limits.

Note: Rolling over a vault store partition does not roll over the vault store database.

Typically the Vault Stores that may require rolling over will be journal based Vault Stores which have a very high turnover of archived items. The approach to rolling over journal based Vault Stores would be to create a new Vault Store and new archives then change the Enterprise Vault journal tasks to use the new archives.

Note: You cannot consolidate Vault Store databases.

Chapter 6

Monitoring

Regular monitoring will enable a baseline performance profile to be measured and used for comparison over time to identify any trends in performance.

As of SQL Server 2005 the Data Collector can be used to gather and store performance metrics in order to monitor trends over time. The SQL Server 2005 Data Collector can be downloaded from Microsoft, and is integrated into SQL Server 2008 onwards. Dynamic Management Views can also be used to provide current performance metrics.

If previous versions of SQL Server are still in use, Windows System Monitor will need to be used to gather performance metrics to log files for subsequent analysis.

Monitor the Enterprise Vault SQL servers during particular activities to ensure that the environment is performing correctly and allow appropriate database tuning.

Example activities to monitor for benchmark purposes are the following:

- During daytime activities which will include end-user workloads
- During mailbox archiving windows
- During journaling, typically daytime
- Overnight during activities such as mailbox or folder synchronization

Remember that, in isolation, these activities do not represent peak load. In production use, the database is also under load from combined activities including ad-hoc administrator actions.

You can use Windows System Monitor to obtain system and SQL Server statistics. Typically, you should monitor the following counters.

Object	Counters	Instances
PhysicalDisk (and potentially LogicalDisk as well)	Avg. Disk Read Queue Length Avg. Disk Write Queue Length Disk Transfers/sec	SQL Data and log file drives

Object	Counters	Instances
	Avg. Disk Bytes/Transfer	
	Avg. Disk sec/Transfer	
	Disk Bytes/sec	
	Split IO/sec	
Memory	Page Faults/sec	
	Pages/sec	
	Available Bytes	
Processor	% Processor Time	_Total & All processors
	% Privileged Time	
	% Interrupt Time	
System	Processor Queue Length	
	Context Switches/sec	
SQLServer:Buffer Manager	Buffer cache hit ratio	
	Page life expectancy	
	Procedure cache pages	
	Lazy writes/sec	
	Checkpoint pages/sec	
SQLServer:Access Methods	Page Splits/sec	
	Full Scans/sec	
SQLServer:Memory Manager	Total Server Memory (KB)	
	Target Server Memory (KB)	
SQLServer:Databases	Transactions/sec	_Total & Discovery Accelerator Database
SQLServer:SQL Statistics	Batch Requests/sec	
	SQL Compilations/sec	
	SQL Re-Compilations/sec	
SQLServer:Locks	Average Wait Time(ms)	_Total
	Lock Timeouts/sec	
	Lock Waits/sec	
	Number of Deadlocks/sec	
SQLServer:Latches	Average Latch Wait	

Object	Counters	Instances
	Time(ms)	
	Latch Waits/sec	

CPU and memory

The % Processor Time for the _Total counter indicates overall system activity, but it may be worth monitoring the individual processor counters to see if any particular processors are heavily loaded for sustained periods.

If the % Processor Time is generally above 80%, and the Processor Queue length is generally above twice the number of allocated CPU cores, then the CPUs are likely to be a bottleneck. However, in SQL Server, high CPU use can be an indication of other factors such as ineffective indexes.

In addition, if the context switches/sec are above 15,000 per allocated CPU core when you experience high CPU, it is possible that the server is spending too much time switching between threads of equal priority (but only if the CPU time is above 80%). This may occur for various reasons, as described in the Microsoft books. However, this is most likely to occur with other co-existing software such as multiple SQL Server instances. In this situation, see the section “Multiple SQL Server instances” on page 33.

SQL Server should normally manage memory allocation automatically and avoid situations where memory paging can occur. However, it would be worth monitoring the memory counter Pages/sec, which records the number of hard faults to disk. If there are significant or sustained hard faults, trace the problem to the source. Watching the other SQL Server metrics listed below should also help to indicate if memory capacity is a bottleneck.

Disk

Typically, the disk read/write queue length counters are monitored for evidence of a disk bottleneck. The queues should not normally rise above twice the number of disks in the respective arrays.

Monitor the average disk sec/transfer to measure the latency of the device accesses. Ideally, this should be approximately 5 - 15ms. However, anything in excess of 20ms is of concern.

The use of these counters may not be appropriate when using a SAN, and the hardware vendor’s tools should be used instead.

The Split IO/sec counter can indicate evidence of file fragmentation, and high levels should be addressed with file defragmentation. The remaining counters can be used to measure transfer rates and throughput.

Note: The physical disk counters represent all activity to a physical disk or array, which may contain several partitions (logical drives). The logical disk counters monitor activity to individual logical drives, so they can be used to identify which logical partition is utilizing the disks.

SQL Server

You can monitor the SQL Server performance counters to indicate workload and performance problems. Typically, the following counters should be monitored.

Counter	Notes
Buffer Cache Hit Ratio	Should be above 90% to avoid too much I/O. A lower value may indicate too little server memory.
Total Server Memory, Target Server Memory	If the total server memory is equal to the target server memory, there may be an issue with memory pressure. Examine the other SQL counters to help determine if the server may require more memory.
Page Life Expectancy	Indicates how long pages remain in memory. Values that are regularly less than 300 seconds may indicate insufficient memory.
Lazy Writes/sec	Indicates how many times the lazy writer is moving changed pages to disk to free buffer space. This should be quite low. High values indicate high I/O, which more memory will help to reduce.
Page splits/sec	Ideally should be around or below 80 – 100 per second. Index fill factors can be examined to improve this situation.
Batch Requests/sec, Transactions/sec(_Total)	Can indicate the number of SQL requests, and therefore the overall load the SQL Server is handling.

As well as monitoring the system counters, you can extract more detailed information from SQL Server to identify potential performance issues and enable specific tuning.

You can measure the amount of time that is spent waiting for I/O operations using a SQL Server system table-valued function, fn_virtualfilestats. The following query displays the database files and the average time spent waiting on I/O (both read and write):

```
SELECT file_name(FileId),IoStallMS/(NumberReads+NumberWrites) as 'Avg IO
wait (ms)'
FROM ::fn_virtualfilestats(DB_ID('<database_name>'), -1)
```

Where <database_name> is the name of an Enterprise Vault database

An average value above 20ms suggests that the I/O subsystem could be the source of a bottleneck.

Note: This displays an average since the database was created, and therefore any changes in hardware will not reflect an accurate change in this query. Instead, the IoStallMS column should be measured at intervals over several hours and the deltas used to determine improvements.

It is essential to measure the index fragmentation for particular key tables, as described in “Maintenance”on page 35.

In SQL Server 2005 onwards, you can execute the following SQL statement. This outputs statistics on all tables and indexes where the external fragmentation exceeds 5% and the table consists of at least 1,000 pages.

```
SELECT OBJECT_NAME(i.object_id) AS TableName,
i.name AS TableIndexName, phystat.avg_fragmentation_in_percent
FROM sys.dm_db_index_physical_stats(DB_ID(), NULL, NULL, NULL,
'DETAILED') phystat
JOIN sys.indexes i ON i.object_id = phystat.object_id AND i.index_id =
phystat.index_id
WHERE phystat.avg_fragmentation_in_percent > 5 AND phystat.page_count >
1000
```

Note: This query may take several minutes to complete, depending on the size of the database.

In SQL Server 2000 you can execute the following SQL statement for each table of interest:

```
DBCC SHOWCONTIG (Saveset) WITH ALL_INDEXES
```

From the resulting output, check Logical Scan Fragmentation. This should be as close to 0% as possible, but up to 10% is probably acceptable. Anything higher indicates external fragmentation (pages out of order). Any such indexes that consist of at least 1,000 pages are good candidates for index defragmentation.

The following queries can be used on the Enterprise Vault databases to gather various metrics.

Useful queries

The following queries may be useful in monitoring and tracking Enterprise Vault activity.

Identifying vault store throughput per hour

Execute the following query against each Vault Store database. It outputs the hourly storage throughput rate and both original and compressed data size for the past 24 hours.

```
SELECT "Archived Date" = left (convert (varchar,
archiveddate,20),14),
"Hourly Rate" = count (*),
"MB (orig)" = sum (originalsize)/1024/1024,
"MB (comp)" = sum (itemsize)/1024
from saveset with (nolock) inner join savesetproperty with (nolock)
ON saveset.savesetidentity = savesetproperty.savesetidentity
where archiveddate > dateadd("hh", -24, getdate ())
group by left (convert (varchar, archiveddate,20),14)
order by "Archived Date" desc
```

Note the following:

- The time period can be adjusted by changing the where clause. For example, you could change the following line:

  ```
  where archiveddate > dateadd("hh", -24, getdate ())
  ```

  To a specific date range such as:

```
where archiveddate > '2012-01-01 18:00:00 ' and archiveddate <
'2012-01-01 00:00:00'
```

- Depending upon the size of the database this query may take some time to complete.
- Avoid executing the query during heavy load, particularly during archiving or journal archiving periods.

The following is sample output from the query:

Archived date	Hourly rate	MB (orig)	MB (comp)
2011-10-08 09:	106753	10483	7830
2011-10-08 08:	107506	10470	7857
2011-10-08 07:	107150	10554	7835

Identifying sharing group sharing ratio

Execute the following query against each fingerprint database. It outputs the number of sharable parts stored, the number of references to parts from archived items and the SIS ratio (average references per stored part).

```
DECLARE @sCmd nvarchar(1000)
DECLARE @curTable  int, @LastTable int DECLARE @tbls TABLE (rownum
int IDENTITY(1,1) NOT NULL, TableName nvarchar(255) NOT
NULL,OwnerName nvarchar(255) NOT NULL)
CREATE TABLE #Results (dcount float, drefcount float)
INSERT @tbls (TableName, OwnerName) SELECT object_name(o.id),
u.[name] FROM sysobjects o INNER JOIN sysusers u ON u.uid = o.uid
WHERE o.type='U' and object_name(o.id) like  'MemberTable____'
SET @LastTable = @@ROWCOUNT
SET @curTable = 0
WHILE (@curTable < @LastTable)
BEGIN   SET @curTable = @curTable + 1
            SELECT @sCmd = N'SELECT cast(count(*) as float) as
''dcount'',cast(sum(refcount) as float) as ''drefcount'' FROM ' +
LTRIM(RTRIM(OwnerName)) + N'.' + TableName + N' with (nolock)'
                        FROM @tbls
                        WHERE rownum = @curTable
```

```
            INSERT #Results EXEC sp_executesql @sCmd
END
SELECT SUM(dcount) as 'parts_stored',SUM(drefcount) as
'part_references',SUM(drefcount)/SUM(dcount) as 'SIS_ratio' FROM
#Results
DROP TABLE #Results
GO
```

Note the following:

- Depending upon the size of the database this query may take some time to complete.
- Avoid executing the query during heavy load, particularly during archiving or journal archiving periods.

The following is sample output from the query:

parts_stored	part_references	SIS_ratio
484583	5414621	11.173