Symantec Enterprise Vault™ Performance Guide

Windows

7.0

Symantec Enterprise Vault 7.0 Performance Guide

THIS DOCUMENTATION IS PROVIDED “AS IS” AND ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT SUCH DISCLAIMERS ARE HELD TO BE LEGALLY INVALID, SYMANTEC CORPORATION SHALL NOT BE LIABLE FOR INCIDENTAL OR CONSEQUENTIAL DAMAGES IN CONNECTION WITH THE FURNISHING PERFORMANCE, OR USE OF THIS DOCUMENTATION. THE INFORMATION CONTAINED IN THIS DOCUMENTATION IS SUBJECT TO CHANGE WITHOUT NOTICE.

Symantec Corporation
20330 Stevens Creek Blvd.
Cupertino, CA 95014
www.symantec.com

Contents

Introduction

This document provides guidelines on expected performance when running Symantec Enterprise Vault.

Every customer has different needs and different environments. Many factors influence the performance and needs of an archiving system. These include the type and size of data that is archived, the file types, the distribution lists, and so on. In addition, most systems expect growth in both volume and size of data, and indeed the very existence of an archiving service may lead users to allow their mail or files to be archived rather than delete them. All this leads to the need to be very cautious when sizing a system.

For all the major components of Enterprise Vault 7.0, there is no difference in performance from previous versions of Enterprise Vault. Upgrades from the previous version are quick unless there are many items in the StorageDelete table in the Vault Store database, and therefore this table should be empty or near empty when upgrading.

This guide has a separate section for each of the archiving agents. Each section has an example from testing. These are included to give an idea of where resources are used by the agents so that some performance counters such as disk IOPS and network traffic can be seen on a working system.

Hardware

The most critical factor in the performance of Enterprise Vault is the specification of the system that is used—the servers, the disk systems, the memory, and the network.

Servers and processors

Recommended hardware

There are many processor and server types, and although their speed in GHz may not be an indication of their actual power, this document assumes that your processors are 2.8 GHz or above. In testing, servers with two dual core processors were found to be as powerful as four-processor servers.

In general, the more powerful the processor, the higher the ingest and retrieval rates. However, the other components of the system—the disks, memory, network, and archiving source—need to match this power. For this reason, you should only consider anything more powerful than a four-processor server if your needs demand it.

Recommended memory

In general, between 1 GB and 2 GB memory is sufficient for most day-to-day archiving operations. However, it is best to configure Enterprise Vault with 4 GB of memory to deal with situations when more memory is beneficial, such as conversions of large files or concurrent indexing. There is no benefit to having more than 4 GB of memory on the Enterprise Vault server, and you should not set the /3GB /PAE boot flags.

Hyperthreading

Turning hyperthreading on or off makes no noticeable difference to the overall performance, so we recommend that the manufacturer's default setting is not changed.

Storage

Enterprise vault store

One of the purposes of Enterprise Vault is to allow cheaper storage to be used to archive data. The primary requisite is that the archived data is secure and retrievable.

Symantec works with many storage vendors to certify their systems and ensure that the performance of these systems is acceptable. We do not intend to list these systems here, although EMC Centera is discussed in a separate section. In terms of storage cost savings, there is most benefit in keeping archived data on cheaper network-attached storage (NAS). However, you can also make some saving when keeping archived data on more expensive storage, such as a Storage Area Network (SAN), due to the additional compression and single-instance storage that Enterprise Vault provides.

Most network-attached storage (NAS) devices or Centera offer quick archiving and retrieval while providing space, reliability, and security for archived data. Storage systems from most of the major vendors have been tested for performance and found to be perfectly adequate for storing archived data.

Indexes

The storage required for AltaVista indexes depends on how they are used. If fast concurrent searches are required because Enterprise Vault Discovery or Accelerator products are used then fast storage for the indexes is needed—for example, a SAN or direct attached storage (DAS). On the other hand, if users are prepared to wait for searches then you can use slower systems or NAS.

Note the following about indexes:

- On NAS devices, you must turn off opportunistic locking.
- Indexes will become fragmented whatever the type of device, and this slows down both searching and indexing. You must regularly defragment indexes, ideally while the indexing service is stopped so that defragmentation does not conflict with updates. This is very important if you are using the Accelerator

products. See the “Accelerators” chapter on page 67 for guidelines on how to tune the Accelerator products.

- Enterprise Vault 7.0 is now integrated into Windows Desktop Search, allowing users with Offline Vault to search their own archives without activating the Enterprise Vault indexes.

Local disks

Archiving will generate I/Os on local disks. The primary causes of these are temporary files used when archiving and conversion, and I/Os generated by MSMQ. To separate the I/Os caused by MSMQ, place the MSMQ files on a disk separate from the system disk. The important point is that the local disks do not have to be large, but they should be fast. MSMQ is used for Exchange Archiving and Journaling but not for File System Archiving, Lotus Domino Journaling, PST Migration, or SMTP Archiving.

Network

It is rare that the network is the limiting factor on performance except when some component of the system is on a WAN. For example, there may be a remote Exchange Server or directory database. 100-Base-T is normally sufficient, but see also the sections on network usage for the various archiving agents to calculate what bandwidth is required.

SQL Server

The SQL Server is the heart of the Enterprise Vault system and needs to be properly specified. An SQL Server with the same specification as the Enterprise Vault server normally supports four Vault servers. If the SQL Server is used by other applications, you must take special care to ensure that resources are not taken away from Enterprise Vault.

For best performance, a standard SQL Server with four physical CPUs and a minimum of 4GB of RAM is the recommended specification. You should also tune the performance of the SQL Server using standard methods, such as the following:

- Using the boot flags /3GB and, for systems with more than 4GB of RAM, /PAE.
- Ensuring that the database data files and transaction logs are on different physical striped volumes, with as many disks as possible.
- Using RAID 10 for redundancy rather than RAID 5.

No difference has been seen when comparing SQL 2000 and SQL 2005. SQL 2005 running in 64-bit uses the same resources and does the same amount of work as SQL 2005 running in 32-bit, but it handles locks quicker and is better able to make use of large amounts of memory.

SQL databases need maintenance. As part of the maintenance plan, we suggest that you take the following actions:

- Rebuild indexes
- Update statistics
- Shrink databases

Hyper-threading

Hyper threading may not be beneficial to SQL Server environments and should be carefully tested before enabling.

Upgrading SQL Server

After you upgrade from SQL 2000 to SQL 2005, do the following to ensure that performance is not severely degraded:

```
Exec sp_MSForEachtable 'UPDATE STATISTICS ? WITH FULLSCAN'
Exec sp_MSForEachtable 'ALTER INDEX ALL ON ? REBUILD;'
```

Accelerator database

The Accelerator database server must be well specified and should have a minimum of four CPUs and 4 GB of RAM. (At least 8 GB is recommended, and using x64 architecture can improve the efficiency of the memory utilization.) Split the database log and data files over separate very high speed striped arrays (RAID 10 rather than RAID 5) containing multiple high speed disks and a large array controller cache (512 MB controller cache is recommended for the SQL Server). You can potentially place the log and data files on different SAN partitions. However, the performance has not been measured.

Calculating disk space

This section deals with the space used and how to size for it. When archiving, Enterprise Vault uses three areas of permanent space:

- The Vault Store partition, which is used to hold the DVS (saveset) files. If collections are enabled, they are stored as CAB files. If Centera is the storage medium, it stores the files in its own format.
- The indexes. The current index technology uses AltaVista indexes. Each index is stored in a separate folder and consists of up to 285 files.
- The SQL database, which is used to hold the Directory and Vault Store databases.

The figures that are provided below take all types of archiving into account.

Disk space used by vault stores

When an item is archived, it is first compressed and then metadata is added to it. As a general rule, the item is compressed to half its original size, and the metadata comprises approximately 5 KB. When an item is shared, only the metadata is added.

The following gives some general rules for estimating the amount of storage needed:

1. Halve the total size of items to be archived.
2. Divide by the average number of sharers (typically between 1.2 and 1.5).
3. Add 5 KB multiplied by the total number of items.

The compression ratio may vary considerably. Office documents tend to compress very well. Other document types, such as ZIP and JPEG files, are already compressed and cannot be compressed further. The rule is therefore to overestimate the amount of storage that you need. You must also factor in growth in the number of mailboxes and number and size of messages. Because of these extra factors, a more conservative method of estimating storage is to assume that space used by archiving equals the space used by Exchange in storing items.

The above applies to most types of archiving, but you need to take care with FSA. For example, if FSA is used for compressed image or map files then there is no space saving.

Note: These recommendations do not apply to Centera, which uses a completely different sharing model. See the "Archiving to Centera" chapter on page 63 for more details.

Disk space used by indexes

The space used by indexes is reduced in proportion to the size of the original data as more unindexable data is included in messages or files. To be safe, the recommendations for index sizes have not changed:

- Take the size of the original data.
- Take a percentage of this according to the indexing type.

Indexing type	Percentage
Brief	3%
Medium	8%
Full	12%

However, if all the indexed items are pure text, the index size maybe slightly larger. Conversely, if a high proportion of indexed attachments are non-indexable media or graphics files then the index size may be slightly smaller.

Disk space used by databases

The space used when storing an item in the SQL database has grown in Enterprise Vault 7.0, and it is expected to grow further in future versions. Although the permanent space used by each item is still less than 250 bytes, space has to be allowed for future growth.

In addition to the space used per item, temporary space is used to hold information on items that have not been backed up or indexed. Permanent space is also used to hold data in the Directory database. We suggest that an extra 4 GB is allowed for this.

To calculate the space, proceed as follows:

1. Take the number of items archived.
2. Multiply by 500 bytes.
3. Add 4 GB.

File System Archiving requires more space in the database. To calculate the size of databases when using FSA, proceed as follows:

1. Take the number of items.
2. Multiply by 2500 bytes.
3. Add 4 GB.

There no specific limits on the number of items that you can store in one Vault Store database, but we recommend that a Vault Store does not contain more than 100 million items. This makes it easier to perform database maintenance activities such as rebuilding indexes and backing up databases.

Exchange archiving

In most cases, when you are choosing servers for e-mail archiving, the most critical factor is the ingest rate. For e-mail archiving, there is normally a limited window during the night to archive, and everything must be archived during this period. This window fits in between daily Exchange usage, backups, defragmentation, and all the other background activities that take place on the server.

Choice of CPU

If everything else is going well, it would be a simple matter to select servers with more and faster CPUs to get the maximum ingest rate. In practice, there is usually a limit to the throughput you can achieve from one Enterprise Vault server. Some of the factors that can slow down ingest are as follows:

- Slow local disks
- Heavily used SQL Server
- Overused network links
- Exchange server performance

The following table shows the expected ingest rate for numbers of physical CPUs (assuming 2.8 GHz processors).

Number of CPUs	Ingest rate (70 KB)
1	15,000
2	25,000
4 (including two dual processors)	40,000

Although conservative, these rates cannot be guaranteed.

The average size of mail messages has an effect on the throughput. The observed effect is that when the average message size is doubled, throughput is reduced by one third.

Network usage

You can use the network for the following purposes while ingesting items from Exchange user or journal mailboxes:

- Communicating with and copying data from the Exchange servers.
- Accessing the SQL database.
- Transferring archived data to the storage medium.
- Retrieving archived data from the storage medium for indexing.
- Reading and writing data to and from the index storage medium.
- Background activity: communication with the domain controller, user monitoring, and so on.

Communicating with and copying data from the Exchange servers

Data is compressed before being sent from the Exchange server to the Enterprise Vault server. The rate of compression is dependent on the type of document. As a rule of thumb, assume that the network traffic between the Exchange server and the Enterprise Vault server is equal to the total size of the documents transferred.

Communicating with SQL

A rule of thumb is that 20 KB of total data is transferred between the SQL Server and the Enterprise Vault server for every item archived. If the Directory database is on a different server, 5 KB of this is transferred to the Directory database. More data is transferred to and from the Directory database when empty or sparsely populated mailboxes are archived or when mailboxes have many folders. It is wise therefore to allow at least 10 KB an item when calculating traffic; for example, if deciding whether a Directory database can be sited on a remote server.

Writing to the storage medium

The amount of data being sent and received from the storage medium depends on the single instance ratio and the compression ratio. More data is transferred when the single instance ratio is higher because, when an item is shared, it is read back before being rewritten. In addition, the item is read back for indexing. A rule of thumb is that the network traffic between the Enterprise Vault server and the storage medium is two and a half times the size of the original data.

Reading and writing indexes

When an index is opened, some of the index files are transferred to the Enterprise Vault server. On completion of indexing, the files are written back. Sometimes the files are written back during indexing. The amount of data transferred depends on the number of indexes opened and the size of those indexes. For example, if only one or two items are archived from each mailbox, indexes are constantly opened and closed and a lot of data is transferred, especially if the indexes are large. It is therefore difficult to predict the traffic to the Index server. A rule of thumb is that the network traffic between the Index Storage server and the Enterprise Vault server is equal to the size of the original item for every item indexed.

Summary

For every item archived of average size 100 KB, you can expect the following network traffic.

Enterprise Vault server ↔ Exchange server	110 KB
Enterprise Vault server ↔ SQL Server (Vault Store)	20 KB
Enterprise Vault server ↔ SQL Server (Directory)	10 KB
Enterprise Vault server ↔ Storage medium	200 KB
Enterprise Vault server ↔ Index location	100 KB

Effect on the Exchange server

Most of the time, mailbox archiving is done while mailbox users are not active. There may be occasions when mailbox archiving is needed at the same time as normal Exchange usage. This could be planned to deal with a backlog of archiving or because there is a need to archive during the day.

Enterprise Vault does not take precedence over the active mailbox users. It is not possible to extract items from an Exchange server at the same rate when other mailbox users are active. This is generally good because archiving has less of an impact on active users. If you want to increase the archiving rate at the expense of user's response times or decrease the archiving rate, adjust the number of concurrent connections to the Exchange service used by the archiving process. This is a property of the archiving task.

The effect on the Exchange server can be seen in increased CPU usage and I/Os per second, and in longer response times. This is discussed in the Symantec Yellow Book on Enterprise Messaging Management for Microsoft Exchange (http://www.symantec.com/enterprise/yellowbooks/index.jsp)

Examples

The following examples are taken from testing. These examples are not intended for overall sizing but to give some idea of the effect on different components of the system. Every installation will have different characteristics. In the following examples, the two-CPU Enterprise Vault server is archiving above expectations and the 4-CPU Enterprise Vault server is archiving below expectations. This illustrates first what Enterprise Vault is capable of when all the components are properly specified and tuned, and then how CPU may be underutilized if one component is underspecified. In this case the system disk was hosting MSMQ and had become fragmented.

Enterprise Vault server	**2 CPUs**	**4 CPUs**
Message size	70 KB	70 KB
Archiving rate	38,000	39,000
Overall CPU	99%	60%
IOPS store	74	84
IOPS indexes	5	7
Store KB used/item archived	35	35
Index KB used/Item archived	8	9
Network KB received	912	940
Network KB sent	193	211
Exchange server, average of 2		
Overall CPU	6%	5%
IOPS Exchange store	172	162
Network KB received/sec	72	60
Network KB send/sec	485	408
SQL Server		
Overall CPU	2%	2%
IOPS Total SQL	172	45
Network KB received/sec	98	114
Network KB Sent	105	122

Exchange journaling

Introduction

Exchange Journal archiving and Exchange mailbox archiving act in the same way, and for the most part the same factors that influence the performance of mailbox archiving also influence the performance of journal archiving. There are some differences that make journaling faster than Exchange Mail Archiving. These stem from the fact that only a small number of mailboxes are archived to only a small number of archives.

The main differences are as follows:

- There are fewer connections created and deleted to the Exchange server, and Enterprise Vault archives from one folder only. This leads to more efficient use of Exchange.
- There are fewer calls to the Directory database as permissions are checked less frequently for mailboxes and folders.
- There are fewer indexes opened.
- If archiving from only one journal mailbox to a single journal, only one index server process is indexed. This can lead to indexing falling behind archiving. Where it is important that indexing keeps pace with journaling, it may be necessary to have at least four Journal mailboxes on the Exchange Journaling Server and four archives on the Enterprise Vault server.

Choice of CPU

The following table shows the expected ingest rate for numbers of physical CPUs.

Number of CPUs	Ingest rate (70 KB)
1	15,000
2	25,000
4 (including two dual processors)	40,000

As with mailbox archiving, these figures are conservative, but in some installations even these rates may not be achieved because of the same factors that may slow down ingest from mailboxes. These factors are as follows:

- Slow local disks
- Heavily used SQL Server
- Overused network links
- Exchange server performance

The average size of mail messages has an effect on the throughput. The observed effect is that when the average message size is doubled, throughput is reduced by one third.

Example

This is an example only. It shows how the size of the data influences the ingest rate and other factors.

Enterprise Vault server	1 CPU	2 CPUs	4 CPUs
Message size	32KB	170 KB	321 KB
Ingest rate	74,266	34,496	19,394
Overall CPU	35	45	60
IOPS store	96	57	45
IOPS indexes	5.	6.	8
Store KB/item archived	18	83	204

Index KB/item archived	2.9	8.4	17.6
Network KB received	585	1094	1440
Network KB send	215	125	80
Exchange server, average of 2			
Overall CPU	7%	9%	7%
IOPS Exchange store	113	140	159
Network KB received/sec	267	516	679
Network KB Send/sec	89	57	26
SQL Server			
Overall CPU	5	5	5
IOPS SQL	40	32	25
Network KB received	60	50	37
Network KB send	66	47	31

Tuning parameters for Exchange mailbox and journaling

The rate at which items are archived depends mainly on the specification of the system; that is, the processing power, the IO capacity, and the network. There are some parameters that can be changed for particular problems. It is not suggested that any of the following are used as a matter of course.

Message queues

The main processes when ingesting items are:

- Extracting mail from Exchange.
- Inserting items into the archives.

Items get passed between these processes by message queues. The queues used by mailbox archiving are:

- A1 - Changing pending item to shortcut and errors.
- A2 – Manual archive requests and error handling/retry from A3 queue.
- A3 – “Run Now” requests made from the administration console – one entry for each mailbox to be archived.
- A4 – Error handling/retry from A5 queue.

- A5 – Scheduled archive – one entry for each mailbox to be archived.
- A6 – Mailbox synchronization request.
- Storage Archive – Store item in an archive.

The queues used by Journal Archiving are:

- J1 – Delete message (after successful archive) or change pending item back (on error).
- J2 – Items to process.
- J3 – Mailbox to process.
- J4 – Synchronize mailboxes
- Storage Archive – Store item in an archive

Some of these queues can be used to see whether to adjust the number of concurrent connections to the Exchange server or the number of storage archive processes.

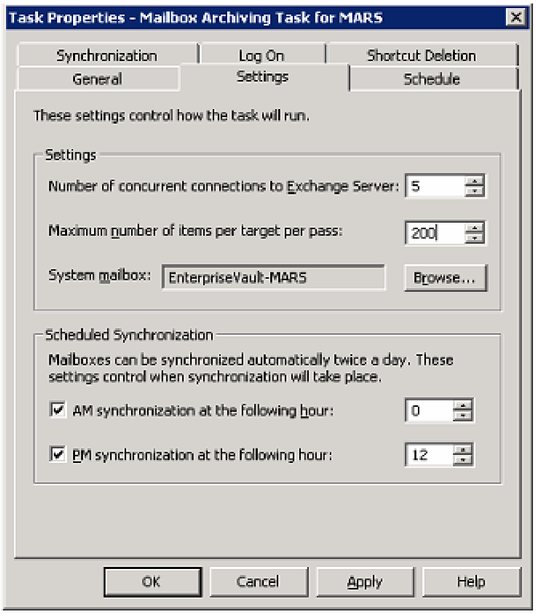
Setting the number of connections to the Exchange server

If you want items to be extracted at a faster rate, you can increase the number of concurrent connections to the Exchange servers. The indications to do this are as follows:

- You are achieving less than the required archiving rate.
- The Storage Archive queue frequently dips to zero.
- For Journal archiving, the J2 queue frequently dips to zero.

If this is the case, increase the number of concurrent connections to 10 from the default of 5.

You can also reduce the number to minimize the impact of archiving on Exchange.



These settings control how the task will run.

Settings

Number of concurrent connections to Exchange Server: 5

Maximum number of items per target per pass: 200

System mailbox: EnterpriseVault-MARS

Scheduled Synchronization

Mailboxes can be synchronized automatically twice a day. These settings control when synchronization will take place.

☑ AM synchronization at the following hour: 0

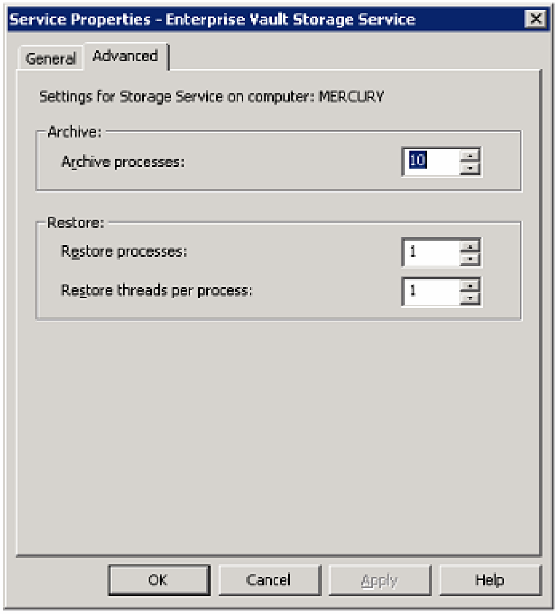
☑ PM synchronization at the following hour: 12

Setting the number of storage archive processes

The default number of archive processes is 5. On more powerful servers it can be advantageous to increase this number. For a four-processor server or a server with two dual core processors, it should be increased as a matter of course. The indications that the archiving rate could benefit from increasing this value are as follows:

- The number of items on the MSMQ queue "enterprise vault storage archive" constantly exceeds the threshold value of 1,500.
- While archiving, the total CPU used by the system averages less than 80%.

The effect of increasing this value is that more processes are available for processing the items on the queue.



Settings for Storage Service on computer: MERCURY

Archive:

Archive processes: 10

Restore:

Restore processes: 1

Restore threads per process: 1

Storage queue threshold limits

When archiving, the Archiving Service takes items from the Exchange server and puts them onto the MSMQ queue "enterprise vault storage archive". In order to prevent an excessive number of items from being put on the queue, a limit is set to the number of items. When this limit is breached, the archiving tasks will pause to give the Storage Archive processes time to clear the queue. By default, this limit is set at 1,500. The number of items may go higher than this because the queue is not checked constantly, so a second higher limit of 4,500 is also set.

By default, when the lower limit is breached, the archiving tasks pause for one minute. When the upper limit is breached, the archiving tasks pause for five minutes. On fast systems, the queues may be cleared in less time with the result that the system remains idle for a few seconds until the minute elapses. The storage processes run faster during this time because they are not competing with the idle Archiving tasks.

The number of items on the Storage Archive Queue can be monitored using Performance Monitor.

```
"MSMQ Queue(evserver\private$\enterprise vault storage archive)\Messages
in Queue"
```

If you see this behavior, raise the following registry values.

Value	Key	Content

`ThrottleLowerThreshold`	`HKEY_LOCAL_MACHINE`
	`\SOFTWARE`
	`\KVS`
	`\Enterprise Vault`
	`\Storage`
		DWORD value set to an integer value. Default is 1500. Raise this to 5000.

`ThrottleUpperThreshold`	`HKEY_LOCAL_MACHINE`
	`\SOFTWARE`
	`\KVS`
	`\Enterprise Vault`
	`\Storage`
		DWORD value set to an integer value. Default is 4500. Raise this to 10000.

Converting to HTML or text

By default, items are converted to HTML. This provides text suitable for indexing and allows a formatted display when items are read in HTML from Archive Explorer. The original item is also stored, and this is what is displayed when downloading an item — for example, by opening a shortcut or viewing an item from the integrated browser.

It is more expensive to convert items to HTML than to text, and you can minimize CPU usage by converting some or all items to text. For Word documents and Excel documents there are special registry keys that you can change to force this, and for all file types there is a registry key. The general registry key can also be used for Word and Excel.

Because of the expense of converting, Excel files are converted to text by default. It is also expensive in CPU usage when converting PDF files to HTML, and this might be considered as a candidate for converting to text.

Value	Key	Content

`ConvertWordToText`	`HKEY_CURRENT_USER`
	`\SOFTWARE`
	`\KVS`
	`\Enterprise Vault`
		String value. 0 (default) - Convert to HTML, 1 - Convert to text.

`ConvertExcelToText`	`HKEY_CURRENT_USER`
	`\SOFTWARE`
	`\KVS`
	`\Enterprise Vault`
		String value. 0 - Convert to HTML, 1 (default) - Convert to text.

Value	Key	Content

`TextConversionFileTypes`	`HKEY_CURRENT_USER`
	`\SOFTWARE`
	`\KVS`
	`\Enterprise Vault`
		String value containing list of file types.
		The list format is:
		`.filetype[.filetype].`
		Each file type must be prefixed by a period and the list ending with a period. For example:
		`.DOC.XLS.PDF`
		All file types can be converted to text by using the * wildcard character. For example, value of .*. converts all file types to text.

Excluding files from conversion

In order to be indexed, items that are not already text must be converted to text or HTML. Some files are excluded from conversion because they contain no textual content, such as JPG files. The most expensive file type to be converted is ZIP because the contents must first be unzipped and then individually converted.

Unknown file types are opened and the first few characters are checked for textual content. Some files may look like text files because they contain valid characters but should not be treated as such and should be specifically excluded. One consequence of not excluding them is that the index may become full of meaningless words.

Value	Key	Content

`ExcludedFileTypes FromConversion`	`HKEY_CURRENT_USER`
	`\SOFTWARE`
	`\KVS`
	`\Enterprise Vault`
		String value containing list of file types. The list format is:
		`.filetype[.filetype].`
		Prefix each file type with a period and end the list with a period. For example:
		`.GIF.JPG.`

Conversion timeout

Large and complex items can take a long time to convert and slow down the whole system during conversion. To prevent such conversions running forever and preventing other work, there is a conversion timeout mechanism. All conversions are abandoned after 10 minutes. Items are normally converted in a fraction of a second, but if conversions are constantly being abandoned—this is an event in the event log—this time can be reduced so that the conversions are abandoned earlier and waste less time. Reducing the time may mean that some items do not have their content indexed; the metadata is still indexed and the item archived as normal.

Value	Key	Content

`ConversionTimeout`	`HKEY_CURRENT_USER`
	`\SOFTWARE`
	`\KVS`
	`\Enterprise Vault`
		DWORD value. Default: 10 (minutes)

Changing the distribution list cache size

Enterprise Vault caches distribution lists and will hold up to 50 distribution lists in cache. In large organizations it is likely that there are over 50 lists. To keep distribution lists in cache longer, the following registry value can be changed.

Value	Key	Content

`DLCacheSize`	`HKEY_CURRENT_USER`
	`\SOFTWARE`
	`\KVS`
	`\Enterprise Vault`
	`\Agents`
		DWORD value set to an integer value. Default is 50 Raise this to 400.

File System Archiving

The basic ingest rate of File System Archiving is much the same as that of e-mail archiving. There is a difference in that MSMQ is not used to hold messages, and this releases some resources. But these extra resources are quickly consumed by the larger size of files. The size and type of archived files have a major effect on the ingest rate.

Number of file servers per Enterprise Vault server

File System Archiving does not impose a heavy CPU load on the file server, but there may be some I/O to the disks containing the files to be archived. To make full use of more powerful servers, it is suggested that there at least two file servers for each Enterprise Vault server, each with a separate File System Archiving task.

Choice of CPU

The choice of CPU depends on three factors:

- The ingest rate
- The file size
- The file type

For general sizing, the following ingest rates should be assumed.

Number of CPUs	Ingest rate (100 KB)
1	15,000
2	25,000
4 (including dual processors)	40,000

The following table shows the ingest rate for some possible scenarios. These figures are for files on an NTFS device and include replacing the files with placeholders. A similar throughput is seen when the files are replaced with placeholders on a NAS device (NetApp). The example shows how the file type influences the throughput rate. For example:

- Text files require no conversion, but large text files do contain more indexable content than other file types.
- Mixed office (Word, Excel, PDF) requires some indexing and conversion.
- PDF files are expensive to convert. A file system consisting just of these will run slower. Some of the expense can be removed by converting PDF files to text rather than HTML.
- A large volume of data can be archived when the files are of type JPG or are similarly unconvertible. Conversion is omitted, and there is only limited indexing.
- The final row (deferred indexing) shows the ingest rate when files are not converted to text or indexed. This is an option available if items are only to be accessed via placeholders and not by any search mechanism. Indexes can be created later but the files have to be converted before indexing. This could take as long as the time to archive the items in the first place.

Document size	Document type	Hourly ingest rate (files)	Hourly ingest rate (MB)
70 KB 2 CPU	Text	44438	3110
100 KB 2 CPU	Mixed Office	46798	4570
200 KB 2 CPU	Mixed Office	28085	5650
540 KB 2 CPU	Big PDF	3243	1710
3013 KB 2 CPU	JPG only	7743	22782
70 KB 4 CPU	Text	57769	4043
100 KB 4 CPU	Mixed Office	60837	5941
200 KB 4 CPU	Mixed Office	46809	9417

Document size	Document type	Hourly ingest rate (files)	Hourly ingest rate (MB)
540 KB 4 CPU	Big PDF	5406	2851
3013 KB 4 CPU	JPG only	12905	37971
100 KB 2 CPU (deferred indexing)	Mixed Office	95906	9365

In addition to the time taken to archive items, time must be allowed for trawling through files on the file server, checking and ignoring items that have already been turned into placeholders or are not eligible for archiving. The following table gives the items trawled when the items are on a single partition on a RAID 5 array. The number of items trawled depends principally on the I/O capacity of the file server and not on the processing power of the file server or Enterprise Vault server.

Items trawled/hour	Disk IOPS
8,037,039	788

Network usage

The network may be used for the following purposes while ingesting item.

- Communicating with and copying data from the file servers.
- Accessing the SQL database.
- Transferring archived data to the storage medium.
- Retrieving archived data from the storage medium for indexing.
- Reading and writing data to and from the Index Storage medium.
- Background activity — communication with the domain controller, user monitoring, and so on.

Communicating with the file server

A rule of thumb is that the amount of network traffic between the Enterprise Vault server and the file server is the size of the data plus 30%.

Accessing the SQL database

A rule of thumb is to allow 20 KB for every item archived to the Vault Store database and 5 KB to the Directory database.

Transferring data to the storage medium and retrieving for indexing

The amount of data being sent and received from the storage medium depends on the single instance ratio and the compression ratio. A rule of thumb is that the network traffic between the Enterprise Vault server and the storage medium is twice the size of the original data.

Reading and writing indexes

When an index is opened, some of the index files are transferred to the Enterprise Vault server. On completion of indexing, the files are written back. Sometimes the files are written back during indexing. The amount of data transferred depends on the number of indexes opened and the size of those indexes. During file system archiving, there is a separate index for each archive point. For example, if only one or two items are archived from each archive point, indexes are constantly opened and closed and a lot of data is transferred, especially if the indexes are large. It is therefore difficult to predict the traffic to the Index server. A rule of thumb is that the network traffic between the Index Storage server and the Enterprise Vault server is equal to the size of the original item for every item indexed.

Summary

For every item archived of average size 100 KB, the following network traffic can be expected.

Enterprise Vault server ↔ Exchange Server	130 KB
Enterprise Vault server ↔ SQL Server (Vault Store)	20 KB
Enterprise Vault server ↔ SQL Server (Directory)	5 KB
Enterprise Vault server ↔ Storage medium	200 KB
Enterprise Vault server ↔ Indexing location	100 KB

Effect on file server

The effect on the file server when archiving is relatively small. The following is an example of the resources used. This is an example only.

File server (200 KB mixed Office at 32537 items ingested hour)	
Total CPU	2.6
Network Bytes Sent/Sec	2518865
Network Bytes Received/Sec	2402706
Network MB Sent/Hour	8564
Network Mb Rcvd/Hour	8169
IOPS Vault Disk	59
IOPS Indexing Disk	6

File types

There are many file types that should be excluded from archiving or included in archiving. For example, it may be only Office files that need to be archived. Again, large files such as .log files should usually be excluded from archiving to prevent the indexes from being cluttered with information that is not useful.

File System Archiving allows the defining of file types that can be included in or excluded from archiving. By careful selection of file types, you can cut down on what files are archived, speeding up the archived process and preventing the indexes from containing unwanted data. For example the preset “Backup and Tmp” file group deletes temporary files rather than archiving them.

File System Archiving and SMTP Archiving

SMTP Archiving allows files to be diverted into a directory location and then archived by File System Archiving. SMTP mail can written to this location almost as fast as mail can be delivered and be ready for archiving.

The folder structure of this location is Year\Month\Date\Hour. When File System Archiving runs, it creates a thread for each folder that it processes up to a maximum of 10 threads. If the SMTP location is being populated while FSA is running, there may not be enough “hour” folders populated with data to occupy all the threads, and FSA may run slower.

Lotus Domino Server

In most cases, when you are choosing servers for Lotus Domino journaling, the most critical factor is the ingest rate. You need to make sure that the servers are able to journal at the required rate during the day. The ingest rate from Lotus Domino servers is less than from Microsoft Exchange servers.

Number of Enterprise Vault servers per Lotus Domino server

In an ideal world, one Enterprise Vault server can match to either of the following and achieve the archiving rates shown in the tables below:

- Two Lotus Domino servers on a Windows operating system.
- One Lotus Domino server on a UNIX operating system (for example, Sun Microsystems sun4u Sun Fire V210).

However, there is always some activity on the Lotus Domino server that competes with archiving and slows down the ingest rate.

To make full use of more powerful servers, we recommend that there at least four Lotus Domino servers for each Enterprise Vault server.

Choice of CPU

The choice of CPU depends on two main factors: the ingest rate, and the file size.

For general sizing, the following ingest rates should be assumed. Note that this is slower than for Exchange archiving.

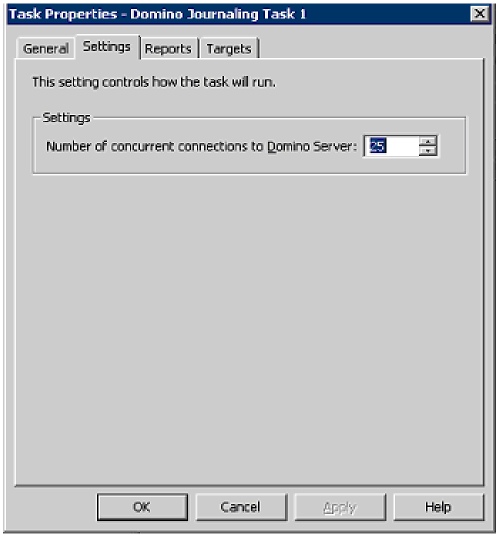
Number of CPUs	Ingest rate (100 KB)
1	10,000
2	15,000
4 (including dual processors)	30,000

Number of concurrent connections to the Domino server

It is important to set up the optimum number of connections for Lotus Domino Archiving. The factors to take into account are as follows:

- What is the desired ingest rate for each Domino server?
- What is the ratio of Domino servers to Enterprise Vault servers?
- What is the impact of archiving on the Domino server?
- Is the Domino server running on Windows?

The number of concurrent connections to the Domino server is set in the Administration Console, where it is a property of the Domino Journaling task. The default is 5 and the maximum is 25.



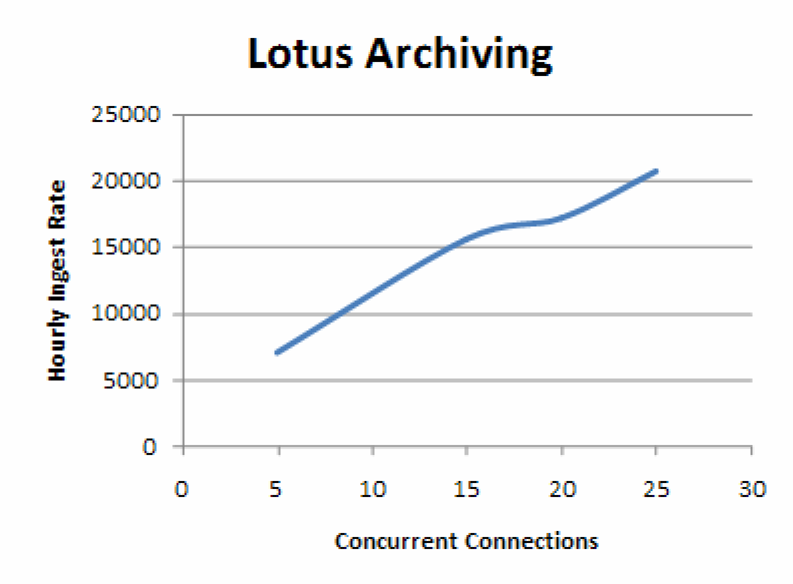
Task Properties - Domino Journaling Task 1

This setting controls how the task will run.

Settings

Number of concurrent connections to Domino Server: 25

The following chart shows how the ingest rate increases as the number of concurrent connections to a single server increases. This is when a Domino server is running a Windows operating system.



The chart shows that when one Enterprise Vault server is archiving from one Domino server running on a Windows operating system, the ingest rate is highest when the number of threads is set at its maximum of 25. This does not mean that the number of connections should automatically be set to 25. The number of Domino servers and the impact of archiving on the Domino servers should also be taken into account as well as the impact on other Domino users.

Ingest rates are higher when the Domino server is on a UNIX-based system. On the system tested, five threads were able to ingest 17,500 items an hour, and 25 threads were able to ingest 35,000 items an hour.

Resources used on the Enterprise Vault server

The following information is for illustrative purposes only. There is no guarantee that these figures will be seen on a working system.

System configuration:

- DELL PowerEdge 2850
- 2 × Xeon 2.8 GHz, Dual Core
- Memory size: 8GB

Example

If the system is set up optimally, the factor that limits ingest rate is the processor power of the Enterprise Vault server. This may not be achievable everywhere; heavily used networks or storage systems may slow down the rates.

The following table gives an example of the resources used.

Hourly ingest rate Av Msg 80 KB	Total CPU	IOPS system disk	IOPS vault disk	IOPS index disk
6979	11.8	13	12	0
15565	34.4	24.1	23.1	0
17147	39.4	25	26	0
20698	48.2	32	30	0
34687	88.9	51.1	50.3	1.6

Memory

Enterprise Vault Lotus Domino Journaling is not normally memory intensive. The following table shows the maximum normal memory usage by the main processes.

Total memory used - 2 tasks	1320
EVLotusDominoJournalTask	220
IndexServer	60

Each Domino task uses 220 MB of physical memory, whereas the index server uses 60 MB. You can comfortably run two tasks with 2 GB of memory.

Any system should be prepared for unexpected memory usage—for example, while converting large documents or searching large indexes while ingesting. For this reason we recommend that any system is configured with 4 GB of memory. There is no advantage to having more than 4 GB of memory.

Resources used by the Domino server (Windows)

This is an example of resources used by the Domino server when Enterprise Vault Journaling but no other activity is concurrent. The ingest rate is slow from one server but the impact on the server is minimal.

System configuration:

- DELL PowerEdge 6650
- 4 × Xeon 2.7GHz
- Memory size: 4 GB

Ingest rate	16,500
Total CPU	2.8
Nserver CPU	2.1
Total IOPS	30

Resources used by the Domino server (UNIX)

This is an example of resources used by the Domino server when Enterprise Vault Journaling but no other activity is concurrent. This is a far less powerful server than the Windows example, but the Enterprise Vault connections are able to make more use of the resources on the Domino server. There is no difference in the resources used on the Enterprise Vault server in journaling the same number of items from a Windows or UNIX server.

System configuration:

- Sun Microsystems sun4u Sun Fire V210
- 1336 MHZ (dual CPU)
- Memory size: 2 GB

Ingest rate	18,000 (5 connections)	35,000 (25 Connections)
Total CPU	21	37
Total I/Os	261	299

Resources used on the SQL server

Ingest Rate	34687
Total CPU	1.1
SQL Server CPU	0.42
IOPS	13.4

The same archives and indexes are used while archiving, so there is little need to refer to the Directory database.

Network traffic

The total network traffic generated by archiving an item of 80 KB is measured as follows. This includes traffic between the Enterprise Vault server, the Domino server, and the SQL Server.

Server	Sent bytes per item	Received bytes per item
Enterprise Vault	242	458
Domino server	535	322
SQL Server	5	14

SharePoint

Introduction

In most cases, when you are choosing servers for SharePoint, the most critical factor is the ingest rate. SharePoint archiving differs from other archiving tasks in that after the initial archiving run, many customers have only a limited number of documents to archive every day.

Number of Enterprise Vault servers per SharePoint server

In an ideal world, one Enterprise Vault server can match to two SharePoint servers on a Windows operating system and achieve the archiving rates shown in the sizing tables below. However, there is a case where Enterprise Vault runs slowly when extracting documents from a SharePoint server. This is where a SharePoint server has thousands of folders, and documents eligible for archiving are sparsely distributed among them. In this case, there is a lower ingest rate but not necessarily more resources used on the Enterprise Vault server or SharePoint server.

Because the ingest rates can be slow from a single SharePoint server, and the resources used on the SharePoint server are higher compared to other archiving targets, we recommend that there at least four SharePoint servers for each Enterprise Vault server.

Choice of CPU

The choice of CPU depends on three factors:

- The ingest rate
- The file size
- The file type

For general sizing, the following ingest rates should be assumed for average document sizes of 200 KB.

Number of CPUs	Ingest rate (200 KB)
1	10,000
2	20,000
4 (including dual processors)	30,000

Resources used on the Enterprise Vault server

The following information shows the resources used when archiving from one SharePoint server, and is for illustrative purposes only. There is no guarantee that these figures will be seen on a working system.

System configuration:

- DELL PowerEdge 2850
- 2 × Xeon 2.8GHz, Dual Core
- Memory size: 8GB

Resources used in Enterprise Vault server

Feature	Value
Ingest rate (items) per hour	15941
Ingest rate (MB) per hour	3270
Total CPU	15.6
IOPS system disk	18
IOPS vault disk	29
IOPS index disk	3

Resources used on the SharePoint server

This is an example of resources used by the SharePoint server when SharePoint archiving is active but no other activity is concurrent.

System configuration:

- DELL PowerEdge 2850
- 2 × Xeon 2.8GHz
- Memory size: 2 GB

Feature	Value
Ingest rate	15941
Total CPU	32.5
SQL CPU	9.1
W3WP CPU	21
IOPS	152

Resources used on the SQL Server

Feature	Value
Ingest rate	15941
Total CPU	0.8
SQL Server CPU	0.5
IOPS	14.4

Network traffic

The total network traffic generated by archiving an item of 200 KB is measured as follows. This includes traffic between the Enterprise Vault server, Domino server and SQL Server.

Server	Sent bytes per item	Received bytes per Item
Enterprise Vault server	28	240
SharePoint server	260	14
SQL Server	13	22

PST migration

Introduction

In general, the rate at which items are migrated into Enterprise Vault from PSTs is the same or better than the rate at which items are archived from Exchange. Large-scale migrations need careful planning because they can take several months or longer and may compete for resources with daily archiving and journaling. It is the planning that is all-important, and it is imperative not to be too ambitious for large-scale PST migrations. You must allow for downtime and for resources to be used by other processes. It is important to ensure that PSTs are fed into the migration process fast enough.

Choice of CPU

There are several methods to migrate PSTs. They all have the same performance characteristics when migrating, except for client-driven migration.

- Wizard-assisted migration.

  PSTs can be migrated into user archives using the Enterprise Vault PST Migration Wizard. This is a quick way of importing a few PSTs, but it is not multi-processed. This means that it is not a suitable method for importing a large number of PSTs in a short time.

  It is possible to run several wizards simultaneously, but this is awkward to set up.

- Scripted migration.

  PSTs can be migrated using EVPM scripts, allowing flexibility over how each PST is processed. This automatically creates five processes when migrating.

- Locate and Migrate.

  In this model, PSTs are located and collected together before migration. It does the work of discovering PSTs on the network before collecting them together into a central area for migration.

- Client-driven migration.

  This is migration initiated by the Enterprise Vault client add-in. This is useful for PSTs on notebooks where the notebook is only intermittently connected to the network. This is the slowest method but has little impact on the client.

The following table shows the archiving rates per hour of items of average size 100 KB for the different migration methods. The throughput figures are when shortcuts are inserted into Exchange mailboxes or PSTs.

Number of CPUs	Wizard-assisted (single process)	Locate and Migrate/ Scripted	Client-driven (single process)
1 CPU	15,000	15,000	2,000
2 CPUs	15,000	25,000	2,000
4 CPUs (including 2 × dual processor)	15,000	40,000	2,000

Location and collection

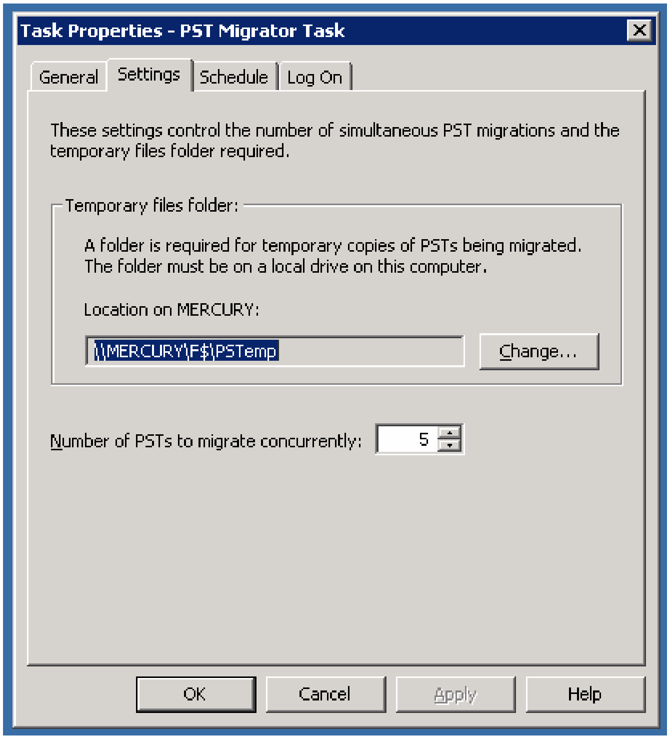
In the Locate and Migrate method, the PSTs are located before migration. The time to locate the PSTs is not predictable and may require many domains and servers to be searched. The following table shows one example.

Number of servers	Number of PSTs located	Elapsed time
40	9000	30 minutes

By default, the PST Collection task copies 100 PSTs to a holding area. As PSTs are migrated, more PSTs are collected to keep the number of PSTs in the holding area constant. This is a continual background process during migration and ensures that there are always files ready for migration.

Increasing the number of concurrent migrations

The default number of concurrent migration tasks is 5. If migration is slower than expected, increase this to 10. If still slower, increase to 15. The number of tasks is a property of the PST Migrator task.



Task Properties - PST Migrator Task

These settings control the number of simultaneous PST migrations and the temporary files folder required.

Temporary files folder:

A folder is required for temporary copies of PSTs being migrated. The folder must be on a local drive on this computer.

Location on MERCURY:

\\MERCURY\F$\PSTemp

Number of PSTs to migrate concurrently: 5

You can also set the number of concurrent tasks in an EVPM script, as follows:

```
ConcurrentMigrations = 10
```

Example

Resources used on the Enterprise Vault server

Ingest rate (items) per hour	Ingest rate (MB) per hour	Total CPU	IOPS vault disk	IOPS index disk
56,895	10056	87.5	130	2

Network traffic

Server	Sent bytes per item	Received bytes per item
Enterprise Vault	11	13
SQL Server	9	11

Search and download

An important part of Enterprise Vault is the user experience when interacting with archived items—for example, searching for items, downloading items, or using Archive Explorer. The user cannot expect the same response times as with the native application, such as Exchange. One goal in Enterprise Vault is to allow the use of slower, cheaper media to store archived items, and retrieval from slower media will always be slower. On the other hand, the user is entitled to expect a reasonable response time, and it is right to set expectations as to when response times are slower.

Types of response time

There are two types of response time:

- The time between the server receiving a request and returning it. This response time is easy to measure and characterize.
- The user-perceived response time between the user action and the requested data being displayed. This is harder to measure and may depend on the user's workstation rather than the server.

Some user interactions require more resources on the server and some on the client server. For example, when retrieving an item, the saveset (DVS) file may be unpacked on the server or client depending on the type of retrieval.

Retrieving items

The first questions to ask are how many items can be downloaded from the Enterprise Vault server, and will this meet the numbers required from a system? All user retrievals invoke an HTTP call to the server. Then either the item is unpacked from the saveset or the saveset is downloaded directly to the server.

There are three main HTTP calls:

- `Properties.asp`.

  This is called when the user downloads an HTML version of the item from the Browser Search.

- `Download.asp`.

  This is called when the user downloads an item from an Outlook shortcut or from the integrated search. This downloads the whole item as a DVS file and unpacks the item. When an item is downloaded from a client earlier than Enterprise Vault 7.0, extra work is done on the server and there is a slower response time. A large file (over 50 MB) could take over a minute to download.

  System managers may choose to download items in MSG format rather than DVS. In this case some more work is done on the server.

- `Viewmessage.asp`.

  This is called when the user downloads an item from ArchiveExplorer or from within a “custom” shortcut (that is, by clicking the hyperlink displayed within the body of the shortcut).

The following table gives an idea of the number of items that can be downloaded. In this case the items were archived on a NearStore R200. As with the other examples in this guide, these figures are for illustrative purposes only. What they show areas as follows:

- For all methods of downloading at a download rate of near 20 items a second, the service time on the server is less than a second. This is shown as IIS time in the chart below.
- The service time has three components: authorization time, IIS time, and time to extract the item with the StorageOnlineOpns process.
- Different download methods return different amounts of data. Download.asp returns the whole item, and so may take longer before the user can view the item. For complex items this may be three or four seconds, depending on the user's workstation. In most cases, however, the response time is less than one second. For other download retrieval methods, the user should see an HTML rendition of the item (but not the item itself) in less than one second.

	Asp type (on dual processor Enterprise Vault server, 2.8 GHz)				
	Properties	**Download DVS – 6.0 client**	**Download DVS – 7.0 client**	**Download MSG**	**ViewMessage**
Concurrent users	6	6	6	6	6
Threads/process (Each process is single threaded for some part of the download.)	10	10	10	10	10
Items/sec	20	18.	27	24	26.
IIS time (millisecs) i.e. average server time for each item.	600	629	362	489	447
Average bytes per download	7391	58832	68767	69343	2934
Vault Server, Dell PowerEdge 2850, 2×XEON 2.85 GHz					
Total CPU%	92.2	98.14	92.8	98.3	79.5
Process CPU out of 400% (dual processor with hyperthreading)					
StorageOnlineOpns	164.7	177.5	70.2	116.3	90.8
w3wp	39.3	75.2	110.7	105.2	51.6
Directory service	9.2	8.7	12.0	11.7	11.3
svchost	83.5	65.1	102.3	89.9	86.5
lsass	21.0	17.0	24.0	21.8	22.1
authserver	18.9	18.0	25.8	18.6	23.1
Network bytes/second Includes download, fetching from NearStore and other traffic	2284500	3155914	4725221	4409018	2579030

SQL Server, HP DL385, 2 × Opteron 2.8 GHz Dual Core					
Total CPU	1.5	0.88	0.6	0.9	0.89
Network bytes	88937	69801	96603	86433	96626
NearStore vault store partition, NetApp Filer R200					
Read bytes sec	1500048	1439311	2162819	1856613	1941919

FSA - opening placeholders

When a placeholder is opened, the File Placeholder service on the file server makes an HTTP call to the Enterprise Vault server, calling download.asp to retrieve the item. The number of files that can be opened in a period is purposely limited in FSA to 20 recalls within 10 seconds. If the limit is exceeded, it is not possible to open placeholder files for a few seconds. You can disable this limit and open at least 20 files every second. The recall process adds approximately 0.25 seconds to the time to open a file.

Searching indexes

There are many factors that determine the time taken to return a search result. When a user searches an index, the index is opened and some of the files are brought into memory. The first search in a series of searches takes longer.

For example, typical response times for a user with 100,000 items archived are as follows:

First search	3 to 10 seconds
Subsequent searches	0.5 to 3 seconds

The longer times are more likely from NAS devices or from larger indexes. In most cases, subsequent search times are less than one second when up to 20 searches are in a search session. See also the "Accelerators" chapter on page 67 for more detailed information on search times.

Archive Explorer

When a folder is expanded in Archive Explorer to display a list of messages or subfolders, an index search is triggered. The retrieved information is cached on the client for 24 hours so that, if the user expands the folder again, there is no interaction with the server unless the user specifically refreshes the folder view.

When a user views an item, the message is downloaded using `viewmessage.asp` or `download.asp` (see “Retrieving items” on page 57).

The result is that Archive Explorer is very responsive and can support a large number of concurrent users.

Archiving to Centera

EMC Centera devices offer a reliable means of archiving data with the added advantage that where replica devices are involved, no backups of archived data are necessary. This allows the archiving window to be extended.

Archiving with and without collections

Enterprise Vault offers two methods of storing items in Centera: with collections and without collections. When items are stored in collections, they are first stored in a temporary area and then collected into a single object and stored on Centera. A collection is up to 100 items or 10 MB of data. Collections are recommended because they result in fewer objects on the Centera. This has several advantages:

- No fall-off in performance as the Centera gets fuller.
- Faster replication.
- Faster deletion of expired items.
- Faster self-healing in the event of a failed disk.

Items for collection are stored on a local disk before collection. This needs to be a fast disk but not large.

Centera sharing model

The way that items are shared or single-instanced with Centera differs from other devices. On other devices it is the whole message or file that is a candidate for sharing. With Centera, attachments are detached from the message and stored in Centera, where Centera identifies them as candidates for sharing. The exact rules are as follows.

A saveset with an uncompressed size of fewer than 100 KB is stored unshared. If the uncompressed size exceeds 100 KB, the saveset is examined for “streams”—indexable items or XML streams such as recipient lists—and attachments that can be separately stored. If none is found, the saveset is stored unshared. If the uncompressed size of any of the streams or attachments exceeds 50 KB, it is stored separately and is eligible for sharing.

This model had the advantage that all attachments are shared even if they are attached to different messages or archived separately by File System Archiving. It also means that there is sharing across vault stores. Small messages are not shared; however even though small messages make up the bulk of messages, messages with large shareable attachments usually make up the bulk of the size. Typically only 3% to 6% of messages have attachments that are of a size eligible for sharing. These attachments make up 80% to 90% of the total size of items archived.

Choice of Enterprise Vault server

There is no substantial difference in performance when archiving to a Centera when compared with other storage media. Using collections does add a small CPU overhead as the collection is an extra process. Refer to the tables for each archiving type for the throughput rate. Likewise, there is little difference in retrieval times when individuals view items or perform bulk retrieval operations.

Centera settings

When archiving to Centera without collections, optimum performance is reached when the number of processes is increased. For example:

Number of storage archive processes	10
Number of PST migrators	20

Centera limits

It is not possible to say with certainty what the maximum ingest rate is because we have not been able to test Centera to its limits; this depends on the number of Enterprise Vault servers. A 20-node Centera (Gen 4) has been tested to an archiving rate of at least 250,000 items an hour with and without collections driven by 6 Enterprise Vault servers. The maximum figure is expected to be much higher than this.

NTFS to Centera migration

Items can be migrated to Centera at high rate. The following table shows a typical example when migrating to a 20-node Centera (Gen 3).

Metric	Hourly rate
Saveset files migrated per hour	137,381
GB (original size) migrated per hour	9.4
Enterprise Vault server CPU (2 × processor)	82%
SQL Server CPU (2 × processor)	7%
Centera Storage nodes Av CPU	2%
Centera access nodes	23%

Accelerators

Tuning for the Accelerator products differs in many ways from the rest of Enterprise Vault. It is the area that benefits most from active tuning, and you must consider the following recommendations for every installation.

Accelerator search performance

Factors influencing performance

So many factors influence the search throughput that it is not possible to estimate the expected throughput. The factors that can have an influence are as follows:

- Index server I/O subsystem performance.
- Index service Windows system cache or external file cache (for example, SAN cache).
- Index file fragmentation.
- Index structure: attachment or item granularity.
- Index size: number and size of files, index locations (roughly related to number of unique words), and number of documents.
- Index compaction status (depends upon update frequency).
- Index server CPU availability.
- Index server memory availability.
- Query complexity.
- Volume of results from search.
- Distribution of indexes over index services.
- Single index search frequency.
- Number of indexes searched.
- Database server memory.
- Database server I/O subsystem performance.

- Database server CPU capabilities.
- tblSearchItems size and condition (index fragmentation).
- tblIntDiscoveredItems size and condition (index fragmentation).
- Total number of search results worker threads.
- Search results thread checking period.
- Concurrent user activity such as searching, accepting, or rejecting searches.

The overall search performance is a combination of the related index service performance and the Accelerator result processing that occurs at the database server. An Accelerator search proceeds through three phases:

1. The index is searched.
2. The results are returned to the Accelerator server and inserted into the database in XML form.
3. A database-stored procedure processes the XML to insert rows into the search result table, ready to be accepted or rejected.

Typically, searching a single index for fewer than 50,000 results using a simple, single-term, text-based query may perform with an overall throughput of 700 items per second. The throughput is likely to be less when more than 50,000 results are produced per index. The maximum throughput possible is 1,000 items per second. When you search multiple smaller vaults for volumes returning fewer than 1,000 results per vault, you can search approximately 1.5 vaults per second (depending upon the index server I/O capabilities).

The complexity of the query can impact the search times, but specifying date ranges in particular helps to reduce the scope. The date range can prevent every index being opened and narrows the search within an index. Searching for messages of different sizes within an index does not influence the search times.

The Enterprise Vault indexing service has a very high I/O requirement. A search of a single index can result in a sustained disk activity of 2,000 I/Os per second. Any bottleneck in the I/O subsystem, whether a slow controller and disks, network-based storage, or third-party software that scans file accesses (such as anti-virus) will significantly impact the performance.

The high I/O requirement can slow the overall performance when concurrently searching multiple indexes on the same storage device. This can be helped by reducing the number of searches started per indexing service. If the indexes are distributed over many indexing services then the load is distributed, helping increase the overall throughput. However, if all the indexes are stored on a single storage device, the storage device may become a significant bottleneck.

The Enterprise Vault indexes can be significantly affected by file fragmentation unless regularly defragmented. Delays in terms of minutes per index per 50,000 result set can occur. Index file fragmentation occurs in all situations, and this problem must be managed with a defragmentation maintenance plan on all types of storage device.

The size of the indexes can influence the search time. High volumes of internal index locations can influence the search throughput, although the index files will be very big and therefore high levels of I/O will also be slowing the search.

A single search can consume 100% of a CPU at the index server for the duration of the search, and therefore benefits from multiple processors. Concurrent searches could potentially consume all CPU resources, but this depends on the size of the indexes and the I/O capabilities of the index location storage device, which quickly becomes a bottleneck. At the same time the index server memory can be consumed by complex queries, and concurrent searches and could start to use page file, adding to the I/O overhead. The indexing service should have up to 4 GB of RAM installed.

When the results are returned to the Accelerator server, the processing can consume 100% of a single CPU and consume 225 MB of RAM processing result chunks of 50,000 results. Concurrently searching multiple indexes can consume high levels of CPU (on depending how quickly the index servers are able to produce results) and on average 1 GB of RAM.

The Accelerator databases require regular maintenance to ensure the throughput is maintained. A number of indices should be re-indexed on a regular basis to reduce fragmentation and ensure that the I/O does not become a bottleneck. The Accelerator server benefits from multiple processors (minimum of two) and up to 4 GB of RAM.

Database tuning

You must create a maintenance plan to re-index tblSearchItems and tblIntDiscoveredItems on a regular basis. In addition, the fill factors for indices in tblIntDiscoveredItems may need to be reduced to help improve acceptance performance.

Accelerator tuning

The Accelerator service machines will benefit from multiple processors (minimum of two, but four is recommended) and up to 4 GB of RAM. The recommended Accelerator search thread tuning should help prevent CPU and memory overload.

The Accelerators will search up to 10 indexes per index server concurrently. However, this can overload those index services, consuming CPU, memory, and very high levels of I/O, and significantly degrading the performance as the search concurrency increases. You may need to reduce the number of search threads to balance the load evenly. The optimal value for this setting depends on the I/O subsystem of the index locations. To determine the best value, make several benchmark searches of large indexes and repeat with different numbers of threads. Start with a value of two threads and monitor if the overall throughput changes as the value is changed. Reduce the value of the System Configuration setting **Number of Vault Search Threads** from 10 to 2.

Index service tuning

An index search requires a sustained high level of I/O during the search and potentially repeatedly if the search results in more than 50,000 items. Therefore, the indexes must be placed on a high performance device, avoiding any bottlenecks between the index process and storage medium. Any bottleneck in the I/O subsystem, whether a slow controller and disks, network-based storage, or third-party software that scans file accesses (such as anti-virus) will significantly impact the performance.

The indexing service machines will benefit from multiple processors (minimum of two) and up to 4 GB of RAM. The recommended Accelerator search thread tuning should help prevent CPU, memory, and I/O overload.

If anti-virus software has been installed to scan file and network accesses, this feature should be disabled on the index servers. Support advises that any anti-virus software should exclude the index locations from its file scan due to known issues with anti-virus software corrupting indexes.

The index files quickly become fragmented on disk, even if there is a large volume of free storage capacity. This file fragmentation can cause severe performance problems, which you need to manage on any index storage device. Use either an automated background file defragmentation product or scheduled device defragmentation.

The item granularity index schema is known to perform better than the default index structure of attachment granularity. The current support advice is to change the index structure by adding the registry key below. However, any existing indexes will need to be rebuilt, which can be very time consuming and should be approached in a phased manner.

Value	Key	Content

`SchemaType`	`HKEY_LOCAL_MACHINE`
	`\SOFTWARE`
	`\KVS`
	`\Enterprise Vault`
	`\Indexing`
		DWORD value. Set to 1.

The opportunistic file locking mechanism is known to cause problems when storing index files on NAS storage devices. Therefore, the current support advice is to disable opportunistic locking at the NAS head.

Each search causes an index to open a large number of files, which is known to cause a problem when a high number of concurrent searches are running and system memory is low. This is due to running out of system page table entries, which are located within the kernel address space. The number of entries available depends on how much space is available to the kernel, which is normally up to 2 GB. However, using the /3GB boot switch reduces the kernel space to 1 GB (reducing the space available for PTEs and other system buffers). The /3GB switch allows an individual process to address up to 3 GB of memory rather than 2 GB (but an application must be compiled to use this extra memory and therefore an Enterprise Vault system should not have the /3GB switch enabled). In addition, the use of the /PAE boot switch, which allows access to memory above the normal 32-bit 4GB limit, halves the number of PTEs because two PTEs are now required to address the memory. However, an application must use the Address Windowing Extensions APIs to access that memory, so again is of little use on an Enterprise Vault server and should not be enabled.

Note: These switches may be applicable to other non-Enterprise Vault servers in the environment, such as the SQL Server.

Export performance

Introduction

Several factors can affect the export throughput:

- The I/O performance of the Accelerator export location storage
- The volume of the exported data
- The size of the exported data
- The export format - original format or PST
- The performance of the Enterprise Vault storage services and their storage devices
- Pre-fetch cache setting
- Age and size of data (whether within the pre-fetch cache)
- Pre-fetch cache location I/O performance

The export is I/O bound, so any bottleneck in the Accelerator service export location, Enterprise Vault Storage service, and SQL Server impacts the throughput.

When exporting to the original format, the export can perform in a consistent linear manner without degrading as the volume of messages increases. The size of each exported message impacts the throughput, and larger messages reduce the overall message throughput, depending on the export location I/O capabilities. In a well- specified environment, the system hits the maximum bandwidth above a 60 KB average message size. For example, an export with an average message size of 20 KB might export messages at 211,000 messages per hour or 1.14 MB per second. However, beyond an average size of 60 KB the throughput is limited to 2.6 MB per second, resulting in a throughput falling quickly from 160,000 messages per second.

Exporting to PSTs passes through two distinct phases: the messages are first exported to original format at the rate described above, and then collected into PST files. The PST collection throughput can be impacted by both the volume and size of exported data. This particular phase is very I/O intensive at the Accelerator export location due to reading and writing large volumes of data to the same logical partition.

When the export is running at its best throughput, it consumes the majority of available CPU at the Accelerator server, which may reduce availability for other users such as interactive Web users. The Accelerator export can also increase the CPU utilization at the Enterprise Vault storage services depending upon storage facilities, typically requiring 100% of two CPUs.

Each export that is started is allocated 25 retrieval threads to obtain the messages from the Storage service and place them at the export location. Concurrent exports can result in degraded performance due to the Enterprise Vault Storage service, Accelerator service CPU, or I/O capabilities.

The export throughput can be improved with an adjustment in production threads but is limited by the I/O performance and CPU. The thread adjustment enables the impact of an export on the Accelerator server to be reduced and to provide a more consistent throughput.

An increase in threads beyond 10 threads does not typically appear to increase throughput due to contention at the export location. The default of 25 threads may be too high for a typical environment, and perhaps should be lowered to 15 threads and tuned according to Accelerator export location disk performance, storage service performance, and export priority.

The overall throughput of concurrent exports is equivalent to a single export with the same total number of threads. Therefore, two exports with five threads provide the same overall throughput as one export with 10 threads. The overall throughput is shared between the exports, so each individual export takes longer to complete than if it were run on its own.

When exports are concurrently generating PSTs, the overall throughput becomes more limited by the I/O capabilities of the export location.

If the pre-fetch cache is enabled the export will obtain data from the cache if possible rather than the storage service. This may improve performance if the storage service is unable to provide items at a good rate. However this will increase the I/O load at the accelerator server and could become a bottleneck if the pre-fetch cache is located on the same partition as the export location.

The Accelerator database server must be well specified and should have a minimum of four CPUs and 4GB of RAM. (At least 8GB is recommended, and using x64 architecture can improve the efficiency of the memory utilization.) Split the database log and data files over separate very high speed striped arrays (RAID 10 rather than RAID 5) containing multiple high speed disks and a large array controller cache (512MB controller cache is recommended for the SQL Server). You can potentially place the log and data files on different SAN partitions. However, the performance has not been measured.

Database tuning

A maintenance plan needs to be created to re-index tblIntDiscoveredItems on a regular basis.

Accelerator tuning

The Accelerator service machines will benefit from multiple processors (minimum of two, but four is recommended) and up to 4 GB RAM. The recommended Accelerator thread tuning should help prevent CPU and memory overload.

Tune the number of production threads according to the disk performance, background priority, and expected export concurrency. Change the number of threads for the following circumstances:

1. To ensure that the disks can keep up with the throughput. Otherwise, the unnecessary extra thread overhead and IO load will unnecessarily consume CPU and slow the overall throughput.
2. To ensure better coexistence with interactive Web users by reducing export CPU utilization.
3. To assist when more than one export is usually run concurrently (threads identified in point 1 divided by the number of normally concurrent exports).

You can tune the number of threads by changing the value of the System Configuration setting **Number of Production Items Thread**. The default is 25.

Storage service tuning

The Enterprise Vault Storage service machines will benefit from multiple processors (minimum of two) and up to 4 GB RAM. The recommended Accelerator thread tuning should help prevent CPU overload.

Web server performance

Preventing errors at peaks of activity

You may need to adjust the following ASP.NET environment settings to prevent "Server too busy" and “Request timed out” errors at peaks of activity. To do this:

1. Locate the .NET `machine.config` file, which is typically in the following folder:

   `C:\WINDOWS\Microsoft.NET\Framework\v2.0...\CONFIG`
2. Open the file in a text editor such as Notepad.
3. Find the `httpRuntime` section.

4. Change the values of the following attributes:

Attribute	Default value	Recommended value
appRequestQueueLimit	100	Maximum number of expected concurrent users
executionTimeout	90	600

5. Save and close the file.
6. Locate the Compliance Accelerator `web.config` files, which are in the `AcceleratorAdminWeb` and `ComplianceWeb` subfolders of the Compliance Accelerator program folder.
7. Open the files in a text editor such as Notepad.
8. Find the `httpRuntime` section.
9. Set the value of the `executionTimeout` attribute to 600.
10. Save and close the files.

Network sockets

You must adjust the Web server TCP/IP socket parameters to provide the ASP.NET environment with adequate network sockets at a sufficient reusable rate. To do this:

1. Locate the following key in the Windows registry:

   ```
   HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\
   Parameters
   ```

2. Update the following values, or create them if they do not already exist:

Name	Type	Value (decimal)
MaxUserPort	DWORD	50000
TCPTimedWaitDelay	DWORD	30

Locations of Accelerator Web server and service

By installing the Accelerator client component (Web server) and Accelerator service on separate servers, you can permit the Web server to support more concurrent users and improve the performance of server tasks.

Monitoring your system

This section gives some suggestions on what to monitor to check that your system is performing as expected.

Using Performance Monitor

Performance Monitor provides useful information on the overall performance of the system and may show up potential bottlenecks.

The following is a suggestion of what to collect for a full monitoring of an Enterprise Vault system. You can use a file containing the following parameters to set up Performance Monitor.

```
<HTML>
<HEAD>
<META HTTP-EQUIV="Content-Type" CONTENT="text/html;" />
<META NAME="GENERATOR" Content="Microsoft System Monitor" />
</HEAD>
<BODY>
<OBJECT ID="DISystemMonitor1" WIDTH="100%" HEIGHT="100%"
CLASSID="CLSID:C4D2D8E0-D1DD-11CE-940F-008029004347">
<PARAM NAME="_Version" VALUE="196611"/>
<PARAM NAME="LogName" VALUE="WinSys_NET"/>
<PARAM NAME="Comment" VALUE="Standard Enterprise Vault monitors"/>
<PARAM NAME="LogType" VALUE="0"/>
<PARAM NAME="CurrentState" VALUE="0"/>
<PARAM NAME="RealTimeDataSource" VALUE="1"/>
<PARAM NAME="LogFileMaxSize" VALUE="-1"/>
<PARAM NAME="DataStoreAttributes" VALUE="34"/>
<PARAM NAME="LogFileBaseName" VALUE="WinSys_NET"/>
<PARAM NAME="LogFileSerialNumber" VALUE="1"/>
<PARAM NAME="LogFileFolder" VALUE="C:\PerfLogs"/>
<PARAM NAME="Sql Log Base Name" VALUE="SQL:!WinSys_NET"/>
<PARAM NAME="LogFileAutoFormat" VALUE="1"/>
```

```
<PARAM NAME="LogFileType" VALUE="2"/>
<PARAM NAME="StartMode" VALUE="0"/>
<PARAM NAME="StopMode" VALUE="0"/>
<PARAM NAME="RestartMode" VALUE="0"/>
<PARAM NAME="LogFileName" VALUE="C:\PerfLogs\WinSys_NET_000001.blg"/>
<PARAM NAME="EOFCommandFile" VALUE=""/>
<PARAM NAME="Counter00001.Path" VALUE="\Cache\*"/>
<PARAM NAME="Counter00002.Path" VALUE="\LogicalDisk(*)\*"/>
<PARAM NAME="Counter00003.Path" VALUE="\Memory\*"/>
<PARAM NAME="Counter00004.Path" VALUE="\MSMQ Queue(*)\*"/>
<PARAM NAME="Counter00005.Path" VALUE="\Network Interface(*)\*"/>
<PARAM NAME="Counter00006.Path" VALUE="\PhysicalDisk(*)\*"/>
<PARAM NAME="Counter00007.Path" VALUE="\Process(*)\*"/>
<PARAM NAME="Counter00008.Path" VALUE="\Processor(*)\*"/>
<PARAM NAME="Counter00009.Path" VALUE="\Server\*"/>
<PARAM NAME="Counter00010.Path" VALUE="\System\*"/>
<PARAM NAME="CounterCount" VALUE="10"/>
<PARAM NAME="UpdateInterval" VALUE="60"/>
<PARAM NAME="SampleIntervalUnitType" VALUE="1"/>
<PARAM NAME="SampleIntervalValue" VALUE="60"/>
</OBJECT>
***********************************************************************
```

The following are counters of particular interest.

- \Processor(_Total)\% Processor Time

  When archiving at its maximum rate, a well-tuned Enterprise Vault server is expected to use between 70% and 100% of the total CPU available. If the value is less than this then there may be a bottleneck elsewhere in the system.

- \Process(*)\% Processor Time

  The system may be using CPU but not getting the expected archiving rate. A process may be consuming more CPU than expected, or an application other than Enterprise Vault may be consuming CPU.

  During archiving, you can expect to see the following processes consuming CPU:

  - The archiving task. The name varies according to type of archiving.
  - StorageArchive.exe, used when mailbox or journal archiving. For other archiving types, the StorageArchive functions are within the Archiving task.
  - DirectoryService.exe. A small amount of CPU is used by the Directory service.
  - IndexServer, used to index.

  - EVConverterSandbox, used to convert items into HTML.
  - StorageCrawler, fetches items from Storage for indexing.

  Some processes are recreated frequently such as EVConverterSandbox. This can lead to misleading results from performance monitor, with spurious very high values.

  In addition, some processes may be active that are concerned with Windows authorization and permission checking and for controlling services.

  - Lsass.exe
  - Crss..exe
  - Svchost.exe

- \LogicalDisk(*)\Avg. Disk Queue Length
  \LogicalDisk(*)\Disk Transfers/sec

  The local disks may be overloaded. The Transfers/sec counter can tell you whether you have a high transfer rate, whereas the Disk Queue counter can tell you whether the disk has problems servicing the requested transfers. In general, the disk queue should not exceed 2 (minus the number of active spindles) for extended periods.

- \LogicalDisk(*)\Free Megabytes

  It is useful to know the rate at which space is being used on disks. For example, by looking at the disk on which indexes or vault stores are held, you can calculate the space taken by each archived item. Very often when the data is not stored on NTFS devices, the NAS device supplies its own Performance Monitor Counters.

- \MSMQ Queue(mercury\private$\enterprise vault storage archive)\Messages in Queue

  When items are archived from an Exchange server, they are retrieved from the Exchange server and placed on the Storage Archive queue. See the "Exchange journaling" chapter on page 25 for a description of the expected values and actions to take.

Using SQL Performance Monitor counters to measure archiving rate

There are no inbuilt performance counters to show the current archiving rate. It is possible to improvise by using a SQL query and posting the result of the query to one of the SQL user-settable counters. The archiving rate can then be viewed with the other Performance Monitor counters.

The following SQL runs continuously, updating the performance counters every minute with the number of archived items and the hourly archiving rate. Edit the name of the vault store. You can run this as a SQL job. The job can fail in SQL 2005 although it will run as SQL query.

```
DECLARE @CCOUNT INT
DECLARE @CCOUNT_LAST INT
DECLARE @CCOUNT_DIFF INT
use evname_of_vaultstore
WHILE 1=1
BEGIN
select @CCOUNT = (select rows from sysindexes where name='pk_saveset')
EXEC sp_user_counter1 @CCOUNT
SET @CCOUNT_DIFF = ((3600 / 60) * (@CCOUNT - @CCOUNT_LAST))
EXEC sp_user_counter2 @CCOUNT_DIFF
SELECT @CCOUNT_LAST = @CCOUNT
WAITFOR delay '00:01:00.00'
END
```

Useful SQL

The following SQL snippets can either be pasted directly in SQLQuery or created as separate files. All should be run in the context of the vault store database.

Archiving rate/hour

This shows for each hour:

- The time.
- The number of items archived in that hour.
- The megabytes archived—original size as reported by MAPI when archiving. The actual original size in Exchange may be larger for very small items.
- The megabytes archived—compressed size.

If only the last 24 hours is required, you can uncomment the commented line.

This can be used to check that the archiving rate is as expected and that the compression rate is as expected.

```
select "Archived Date" = left (convert (varchar, archiveddate,20),14),
"Hourly Rate" = count (*),
"MB (original)" = sum (originalsize)/1024/1024,
"MB (compressed)" = sum (itemsize)/1024
from saveset,savesetproperty
where saveset.savesetidentity = savesetproperty.savesetidentity
--and archiveddate > dateadd("hh", -24, getdate ())
group by
left (convert (varchar, archiveddate,20),14)
order by
"Archived Date"
desc
```

`"OriginalSize"` is a field introduced into Enterprise Vault 7.0. For earlier versions, the following can be used. This has no information on the original size.

```
select "Archived Date" = left (convert (varchar, archiveddate,20),14),
"Hourly Rate" = count (*),
"MB (compressed)" = sum (itemsize)/1024
from saveset
--where archiveddate > dateadd("hh", -24, getdate ())
group by
left (convert (varchar, archiveddate,20),14)
order by
"Archived Date"
desc
```

Archiving rate/minute

It can sometimes be useful to know the archiving rate per minute. For example, suppose that the archiving rate is high for a few minutes and then low for an extended period. This may indicate that the system is capable of achieving high archiving rate, but some other factor such as a lack of items to archive may give the impression of a slow archiving rate.

```
select "Archived Date" = cast (archiveddate as smalldatetime),
"Minute Rate" = count (*),
"MB (original)" = sum (originalsize)/1024/1024,
"MB (compressed)" = sum (itemsize)/1024
from saveset, savesetproperty
where
saveset.savesetidentity = savesetproperty.savesetidentity
-- and archiveddate > dateadd("hh", -24, getdate ())
group by
cast (archiveddate as smalldatetime)
order by
"Archived Date"
Desc
```

For versions earlier than Enterprise Vault 7.0, use the following SQL without the original item size.

```
select "Archived Date" = cast (archiveddate as smalldatetime),
"Minute Rate" = count (*),
"MB (compressed)" = sum (itemsize)/1024
from saveset
--where archiveddate > dateadd("hh", -24, getdate ())
group by
cast (archiveddate as smalldatetime)
order by
"Archived Date"
desc
```

Archived files sizes

It is often useful to know the size and compression ratio of files being archived. This permits an understanding of the pattern of email. For each file size by KB, the following SQL shows the number of files archived at that file size and the total original and archived size. Because of integer truncation, an original size of 0 means a size between 0 KB and 1 KB, and so on. For small files, the actual original sizes may be larger than reported.

```
select
"Count" = count (*),
"Original Size" = originalsize/1024, "Archived Size" =
sum(itemsize)/count(*),
"KB (orig)" = sum (originalsize)/1024,
"KB (Arch)" = sum (itemsize)
from saveset,savesetproperty
where saveset.savesetidentity = savesetproperty.savesetidentity
group by
originalsize/1024
```

Sharing ratio

It is useful to know for sizing purposes what the sharing ratio is and whether it matches expectations. The SQL below shows the following:

- The total number of files archived.
- The total number of files created by archiving after sharing.
- The total size of the original items archived
- The total size of the compressed size.
- The sharing ratio achieved and the expected space saving.

```
select cc = count (*),
OriginalSize = sum (originalsize)/1024/1024,
ArchivedSize = sum (itemsize)/1024,
SharedArchivedSize = sum (itemsize)/1024/count (*)
into #temptable
from saveset, savesetproperty
where
saveset.savesetidentity = savesetproperty.savesetidentity
group by
iddatetime,idchecksumhigh,iduniqueno,idchecksumlow
```

```
order by
cc
desc
select "Total files archived" = sum(cc),
"Number of files after sharing" = count (cc),
"Total Original Size" = sum (OriginalSize),
"Total Archived Size before sharing" = sum (ArchivedSize),
"Total Archived Size after sharing" = sum (SharedArchivedSize),
"Estimated size after archiving" = sum (SharedArchivedSize) + (((count
(*)*5) + (sum (cc)*2))/1024)
from #temptable
drop table #temptable
```

Using Log Parser

Log Parser is a powerful, versatile tool that provides universal query access to text-based data such as log files, XML files and CSV files, as well as key data sources on the Windows operating system such as the event log, registry, file system, and Active Directory. The tool is freely downloadable from the following page on the Microsoft site:

http://www.microsoft.com/downloads/details.aspx?familyid=890cd06b-abf8-4c25-91b2-f8d975cf8c07

Log Parser and IIS logs

The following are examples of using queries to extract information from the IIS logs. IIS logs are created daily in the folder `C:\WINDOWS\system32\LogFiles\W3SVC1`.

Turn on the following settings in IIS before you run this request.

- Method (cs-method)
- URI Stem (cs-uri-stem)
- URI Query (cs-method)
- Bytes Sent (sc-bytes)
- Bytes Received (cs-bytes)
- Time Taken (time-taken)

Example 1 - Getting daily hits for all IIS requests

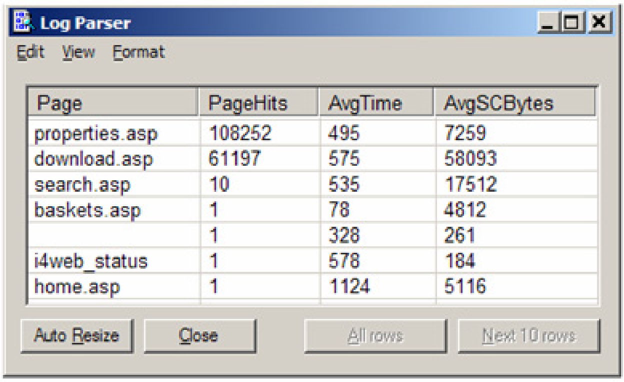
Create a file that contains the following text and name it, for example, `Queries1.sql`. Substitute the name of the log file as necessary.

```
SELECT extract_filename (cs-uri-stem) AS Page, COUNT(*) AS PageHits,
avg (time-taken) as AvgTime,avg(sc-bytes) as AvgSCBytes
FROM
\\MyServer\c$\windows\system32\logfiles\w3svc1\ex061020.log
where sc-status = 200
GROUP BY Page
ORDER BY PageHits DESC
```

Run this file as follows:

```
Logparser file:Queries1.sql -i:iisw3c -o:datagrid
```

Log Parser displays the results as in the following example:



Page	PageHits	AvgTime	AvgSCBytes
properties.asp	108252	495	7259
download.asp	61197	575	58093
search.asp	10	535	17512
baskets.asp	1	78	4812
	1	328	261
i4web_status	1	578	184
home.asp	1	1124	5116

For example, this shows that there were 108,252 downloads using `properties.asp` (called when a user downloads an HTML version from the Web Browser). On average each download took 0.495 seconds, and the average length of the data downloaded was 7,259 bytes.

Example 2: Getting download rates

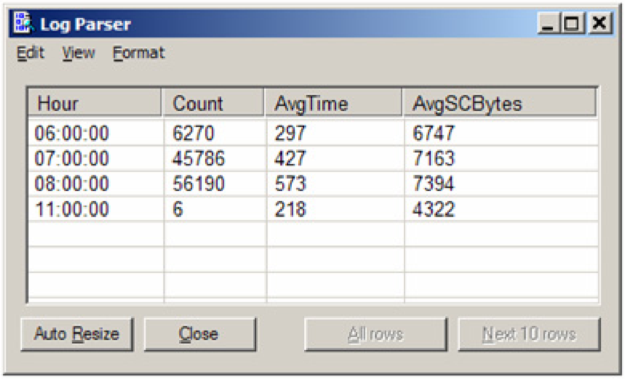
Create a file that contains the following text and name it, for example, `Queries2.sql`. Substitute the name of the log file as necessary.

```
Select QUANTIZE( Time, 3600 ) as Hour, count (*) as Count,
avg (time-taken ) as AvgTime,avg(sc-bytes) as AvgSCBytes
from \\MyServer\c$\windows\system32\logfiles\w3svc1\ex061020.log
where (cs-uri-stem like '%%properties.asp%%')
and sc-status = 200
group by
QUANTIZE( Time, 3600 )
```

Run this file as follows:

```
logparser file:Queries2.sql -i:iisw3c -o:datagrid
```

Log Parser displays the results as in the following example:



Hour	Count	AvgTime	AvgSCBytes
06:00:00	6270	297	6747
07:00:00	45786	427	7163
08:00:00	56190	573	7394
11:00:00	6	218	4322

For example, this shows that between 7 a.m. and 8 a.m. there were 45,786 downloads using `properties.asp`. The average time per download was 0.427 seconds, and the average size 7193 bytes.

Use the results from the first query and substitute the other download or search methods for `properties.asp` in the query. For example:

- `Download.asp` – download from the integrated search
- `Viewmessage.asp` - download from ArchiveExplorer
- `Search.asp` - Browser Search
- `Searcho2k.asp` - Integrated Search

Getting index statistics

You can use Log Parser to obtain indexing statistics from `updates.log`, which is new file in Enterprise Vault 7.0. `Updates.log` is created in the indexing folder for every archive.

Example 3: Getting indexing rates for an archive

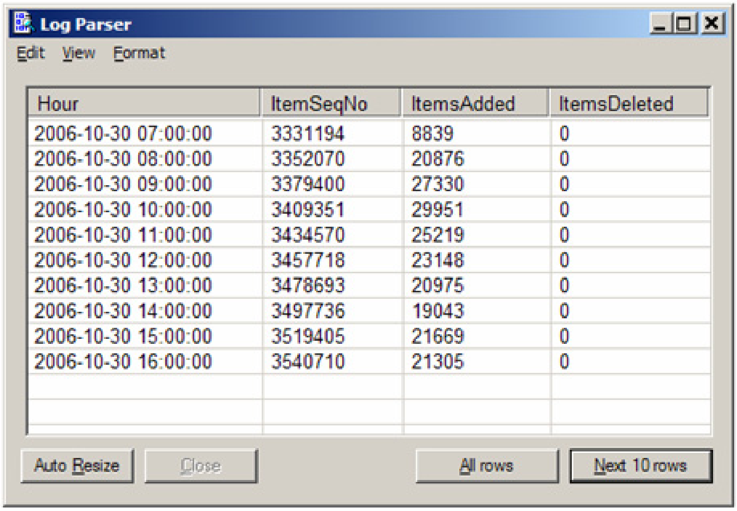
Create a file that contains the following text and name it, for example, `Queries3.sql`. Substitute the name of the log file as necessary.

```
select QUANTIZE( Field1, 3600 ) as Hour,
max(to_int(extract_token(Field5,1,'='))) as ItemSeqNo,
sum (to_int(extract_token(Field7,1,'='))) as ItemsAdded,
sum (to_int(extract_token(Field9,1,'='))) as ItemsDeleted
from
'Indexes\1D8BE8BF16B78D74989E0A47F61E4A40C1110000server.dom.com\updates
.log'
where (Field2 = 'M')
group by Hour
```

Run the query as follows:

```
logparser file:Queries3.sql -i:tsv -o:datagrid -headerrow:off
```

Log Parser displays the results as in the following example:



Hour	ItemSeqNo	ItemsAdded	ItemsDeleted
2006-10-30 07:00:00	3331194	8839	0
2006-10-30 08:00:00	3352070	20876	0
2006-10-30 09:00:00	3379400	27330	0
2006-10-30 10:00:00	3409351	29951	0
2006-10-30 11:00:00	3434570	25219	0
2006-10-30 12:00:00	3457718	23148	0
2006-10-30 13:00:00	3478693	20975	0
2006-10-30 14:00:00	3497736	19043	0
2006-10-30 15:00:00	3519405	21669	0
2006-10-30 16:00:00	3540710	21305	0

This provides the following information for this index for each hour in updates.log:

- The index sequence number
- Items added in the hour
- Items deleted in the hour

Miscellaneous

VMware ESX Server 3.0

We have found this to be an effective way to archive data. There is an overhead when using this virtual server. When archiving, the archiving rate is 65% of the expected rate. This is because there is a CPU overhead when archiving from a virtual server.

The following table shows test results when archiving from a virtual server. Note that these results are for comparison only, and the throughput figures are unlikely to be reached at a customer site. Surprisingly, total throughput was higher when there were two virtual servers on the same physical server and less when there were four.

Server (4 × CPU, 8 GB RAM)	Total items ingested/hour	(Virtual) CPU
Real	66,034	78%
One virtual server, 4 × CPU	43,195	87%
Two virtual servers – 2 × CPU each	55,315	97% (average)
Four virtual servers – 1 × CPU each	41,070	80% (average)

In the above example, the only virtualized server was the Enterprise Vault server. We found that it was possible to have an Enterprise Vault server, Exchange server and SQL Server on the physical server. Although slower at 33,000 items an hour, this is perfectly usable for demonstration purposes.

Collections

After archiving, items may be collected into CAB files of 10 MB, allowing quicker backups or transfer to tertiary storage. The collection process is fairly lightweight and normally runs during the day. Items are collected at approximately twice the rate at which they are archived.

Export to PSTs

On average, items may be exported to PSTs at the rate of 30,000 items an hour. If a higher rate is required, five export processes can be started, giving a maximum rate of 100,000 items an hour.

Storage expiry

Eventually items pass their retention category and must be deleted from Enterprise Vault. Storage Expiry can be concurrent with archiving, but this slows down both archiving and deletion of items. Normally, items are deleted at twice the rate at which they are ingested.

Provisioning

The new Exchange Server Provisioning task completes its work very quickly (normally in one or two minutes). On a site where there are 100,000 users, provisioning takes about 20 minutes.