Enterprise Vault Best Practice Guide

Enterprise Vault 10 Indexing

This document provides design and deployment considerations for the new Indexing engine in Enterprise Vault 10.

If you have any feedback or questions about this document please email them to **EV-TFE-Feedback@symantec.com** stating the document title.

This document applies to the following version(s) of Enterprise Vault: 10.0

Document Control

Contributors

Who	Contribution
Daniel Strydom	Author
Lawrence Grant	Performance content

Revision History

Version	Date	Changes
1.0	August 2011	Initial version
1.1	December 2011	Minor revisions

Related Documents

Version	Date	Title
DA10.0	October 2011	Discovery Accelerator 10 Best Practice Guide
EV10	01/08/2011	Product Installation and Configuration Guide
EV10	01/08/2011	Enterprise Vault 10 Performance Guide

Table of Contents

Appendices

Scope of Document

This document aims to provide guidance on designing and sizing the new 64-bit index engine found in Enterprise Vault 10. The recommendations in this document do not apply to earlier versions of Enterprise Vault.

This document should be used in conjunction with other performance and best practice guides as outlined in the "Related Documents" section of this document.

Intended Audience

This document is aimed at customers, consultants, and support staff. It is assumed that the reader has a thorough understanding of the architecture and operational aspects of Enterprise Vault 10.

Overview of 64-bit Indexing Engine

Enterprise Vault 10 includes a new 64-bit Indexing engine, and the new features and functionality of this engine will need to be considered by both existing customers when upgrading and new customers doing a fresh deployment.

The new 64-bit Indexing engine offers backwards compatibility with existing 32-bit indexes, seamlessly federating searches between old and new indexed data.

In addition to offering backwards compatibility with existing 32-bit indexes, the 64-bit Indexing engine offers the latest in indexing technology, including:

- Index volumes can now each contain up to 5 million items (which is 14x more than EV 9) reducing number of index volumes per archive and improving efficiency

- Concurrent indexing and searching on the same index volume. In previous versions, indexes could not be searched while they were being updated

- Support for up to 2TB RAM, making the most of 64-bit server architecture

- More data in memory allows a greater flexibility in how very large queries and indexes are executed and handled

- Seamless federated searching across legacy 32-bit and new 64-bit indexes, transparent to end users following the upgrade to EV10

- A new set of index management tools have been developed to aid in managing both day-to-day and upgrade scenarios, including workflows with progress and status indicators

- Scalable, distributed architecture enabling customers to size the solution according to their requirements

- The ability to allocate different index hardware and resources to different vault stores so end user and eDiscovery searches don't interfere with each other

- "Closed means closed" 64-bit Index locations allowing making index backups more scalable. No further ingestion or new index volumes but updates/deletes still occur on existing index volumes

- With the recommended hardware specification a typical Enterprise Vault (EV) server can support up to 4x greater user search load, with up to 10x faster search and Archive Explorer results. Discovery Accelerator searches can support up to 3x faster search results and Compliance Accelerator application searches will vary but can support up to 20x faster search results.

EV10 Design Considerations

Server Sizing Considerations

The new indexing engine is more scalable and provides better performance but, as is common with any 64-bit application, it does require additional hardware resources to achieve this. For Enterprise Vault 10 the recommended hardware requirements for an EV server are 8 CPU cores and 16GB of RAM. Customers using this recommended hardware should expect to see equal or better archiving performance when compared to a server running EV9 on the recommended hardware specified for that version. Table 1 shows the new hardware requirements for EV10.

Recommended, if running Indexing service	8 cores, 16GB RAM
Minimum (for low activity sites or servers not running Indexing Service)	4 cores, 8GB RAM

Table 1 - Minimum and Recommended Hardware for EV10

As the load on the EV server CPU has significantly increased, care should be taken to avoid potential issues for oversubscribed virtualized environments and that the recommended minimum hardware specification is met before upgrading.

The memory requirements have also increased, but it will vary considerably depending on the search and ingestion requirements. In a typical environment around 6GB of RAM will be used in core activities increasing to the full available 16GB recommended memory with journaling and searching (end user searches, Virtual

Vault activity, etc.). Adding more memory will improve server performance, for more details on expected server throughput for different CPU and memory configurations refer to the EV10 Performance Guide[1].

64-bit index volumes are loaded and managed by Enterprise Vault to ensure the optimum use of memory, providing better performance for both archiving and searching (whereas 32-bit index volumes are limited to 30 loaded index volumes). In addition there are more processes loaded per 64-bit index volumes than 32-bit index volumes. Therefore you are likely to observe more indexing processes loaded than previously. For example, during archiving you may see perhaps an additional 300 processes in memory than previous versions.

The 64-bit indexing engine requires greater resources and therefore you are likely to observe the EV server utilizing all CPU resource.

Sizing Recommendations for Virtual Machines

The recommendations in this section apply to all EV servers and not just Indexing servers. As a general rule of thumb, running EV on a virtual machine will result in a 25% to 30% reduction in search and archiving throughput performance. This effect can be reduced by properly sizing and configuring the host, ensuring bottlenecks like storage and network I/O does not impact the guest.

The recommended 8 CPU cores can be supported by VMware ESX 4.1 and later virtual machines. Microsoft Hyper-V 2008 R2 virtual machines only support 4 virtual CPU cores; whilst meeting the EV10 minimum server specification 4 cores will only be suitable for small environments or servers only running Storage services (where indexing is offloaded to an separate Index server; see sections later in this document).

The following general recommendations apply to virtualized servers:

- The hypervisor should be Type-1 (native) to ensure the minimum impact on hardware resources
- It is recommended that independent, dedicated data storage locations are used for the guest operating system and data file locations to avoid contention with other guests
- The disk partitions to be used for the index locations should be created as recommended by the hypervisor vendor for random access. In most cases the recommendation is to use virtual hard disks, created as fixed size and not dynamic

---

[1] **www.symantec.com/docs/doc4553**

- The required memory capacity should be dedicated and prioritized to the virtual machine to prevent dynamic allocation or sharing.
- Avoid the use of hyper-threading by the hypervisor
- Set the number of CPU cores to be exclusively dedicated to the virtual machine
- The processor priority and bandwidth should be set to provide the virtual machine with full utilization of the selected CPUs
- See compatibility charts for current supported virtual environments.

Storage Considerations

Careful consideration should be given to storage changes outlined in this section of the document; EV10 has different index sizing and performance characteristics to earlier versions.

In EV10 there are two levels of indexing; Brief and Full. Medium indexing is no longer available.

The storage footprint is also marginally higher than that of the 32-bit engine found in EV9. See Table 2 for Index data footprint (as a % of original unarchived data size).

Index type	Exchange Mailbox Archiving, Journal Archiving and PST Migration	Domino Mailbox Archiving, Journal Archiving and NSF Migration	File System Archiving (Mixed Office documents)	SharePoint Archiving (Mixed Office documents)
Brief	4%	4%	2%	2%
Full	13%	13%	6%	6%

Table 2 – Index data footprint (% of original item size)

After upgrading to EV10, existing medium level index volumes will automatically map to full text index volumes, and will therefore increase from a typical 8% of the original item data size to 13%. Brief level indexes will increase from 3% to 4% for email archiving. Existing EV9 full index volumes will grow from 12% to 13% when upgrading to 64-bit (overall increase of 1% in size).

It's important to consider the type of attachments/documents/files being indexed when calculating the indexing footprint. If you have a large proportion of binary files (images, audio, video, executables) then the indexing footprint will be considerably smaller than the figures given above.

IOPS Requirements for Index Locations

The new index engine places more load on the Index locations. Isolated testing of EV9 showed a typical IOPS load of 50 – 200 during daytime end-user activities, compared to EV10 where the average indexing IOPS is between 175 and 400 for a typical Exchange archiving environment.

- For a typical Exchange environment the IOPS during ingestion measured between 150 and 200

- For a typical File Archiving environment the average indexing IOPS is measured at around 400

The load that Accelerator searches place on the Index volumes will vary depending on the size of the index volume, the number of volumes being searched and the search parameters. The following considerations apply when calculating IOPS for Discovery and Compliance Accelerator:

- For Discovery Accelerator expect between 1,000 and 3,000 IOPS [2] load on the index locations
- Compliance Accelerator is likely to generate a much lower load than Discovery Accelerator because it will generally be searching only the recent index volumes
- If using Accelerators to search Mailbox volumes consider the potential impact the search can have on end user searching, Archive Explorer and metadata updates (either through archiving or Virtual Vault/Vault Cache)
- The storage device for each index server should be able to support the combined load in that scenario, supporting up to 1,500 IOPS.

The IOPS loads for the 64-bit engine are in most cases 3x greater than that of the 32-bit engine, but the 64-bit engine delivers 3x the search throughput.

The load on index locations is calculated on a per server basis. For example, 5 index servers will require either DAS storage providing 400 IOPS for each server, or SAN arrangements providing up to 2,000 IOPS between the 5 servers. Similarly when considering Discovery Accelerator requirements where 5 index servers might require 5,000 to 15,000 IOPS in total.

Choosing a Suitable Storage Device for Index Locations

When choosing a suitable storage device the whole storage solution should be considered. The supported IOPS is just one aspect of performance and other areas should also be looked at such as connectivity – for example Fiber and iSCSI are preferred over CIFS. The storage device should be using the latest firmware, and the server should have the latest device firmware and drivers installed.

As per previous versions of Enterprise Vault, a high speed storage device is recommended for Index locations. Lower tier NAS devices are generally not recommended for index locations, and should definitely not be used to host indexes where any type of Accelerator search is used.

The following table lists a number of environments and scores the suitability of storage devices for each environment (0 being unsuitable, and 5 the best device for that scenario). The scores are based on typical device performance seen in the field, and the assumption is made that the device is optimally configured

---

[2] More information on sizing indexing for Discovery Accelerator is available in the Discovery Accelerator 10 Best Practice Guide available at **http://www.symantec.com/docs/tech159520**

	DAS[3]	SAN	NAS (CIFS)	NAS (iSCSI)
Smaller environments up to 500 users, not using DA	❹	❺	❷	❹
Smaller environments up to 500 users, using DA	❸	❺	⓿	❸
1,000 – 2,500 users per EV server, not using DA	❹	❺	❷	❹
1,000 – 2,500 users per EV server, using DA	❹	❺	⓿	❸
2,500 – 5,000 users per EV server, not using DA	❸	❺	❶	❸
2,500 – 5,000 users per EV server, using DA	❷	❺	⓿	❷

Table 3 - Index Storage Suitability Guide

---

[3] Minimum RAID 1, 1+0 or 6

NAS devices connected over CIFS shares is not suitable to host index volumes for certain environments due to the slower connectivity speed.

If your current environment scores a 2 or less it is recommended that you consider modifying your storage infrastructure to get the most out of Enterprise Vault and protect your indexes from corruption and costly rebuilds.

The best devices are local storage, direct attached storage, or SAN LUNs.

The indexing service file locations require good random access or a high number of IOPS combined with good transfer rates. Therefore, it is recommended that:

- A striped array of many high-speed disks be used
- Use hardware based RAID and not software
- Ideally each indexing service should use multiple locations to distribute the load, located on different arrays to ensure that throughput can be maintained
- To achieve redundancy on the striped arrays while maintaining performance, consider the RAID scheme carefully. RAID levels 5 and 6 are popular and cost-effective methods of achieving redundancy while maintaining striped disk read performance
- A poorly sized RAID-5/6 implementation may reduce the throughput of index updates, creating a backlog. Correctly sizing a RAID-5/6 implementation to maintain write performance may become more costly than RAID-10, and therefore a RAID-10 scheme should be considered.

In the case of local or direct attached storage:

- Use multiple controllers supporting multiple channels to distribute the load between index file locations and provide sufficient throughput
- Provide battery-backed read and write cache to aid performance.

Before using partitions on a SAN, consider the I/O load together with any other applications that are already using the SAN to ensure that the performance can be maintained. Ideally, the implementation should be discussed with the SAN hardware vendor to ensure that optimum performance is achieved. Typically LUNs should be created across as many suitable disks as possible, using entire disks rather than partial disks to prevent multiple I/O-intensive applications from using the same disks.

When creating a basic NTFS volume on the storage device, align the partition with the device track sectors to prevent unnecessary disk operations. You can achieve this through the Windows 2008 command line tool "diskpart". See the Windows Help or Microsoft TechNet for more information on using the tool.

IOPS Requirements for Server Cache Area

As a typical rule of thumb expect 150 – 400 IOPS per index server for the server cache area during normal daytime end-user activities. In large Virtual Vault/Vault Cache environments the server cache area should be able to support up to 1,000 IOPS to cope with frequent client update requests.

In EV10 the server cache area is also used to contain the index results when processing searches where 1,000 or more search results are retrieved, therefore the performance of the server cache area (on servers hosting Indexes) will affect the speed at which Accelerator searches are performed. Ensure the server cache area is able to support minimum 1,000 IOPS, and that a dedicated disk/SAN LUN is used to provide the best performance.

It is recommended that the server cache area be at least 20GB in size. Approximately 35MB of disk space is consumed by 50,000 results.

The use of the Windows temporary folder has not changed in EV10; it remains a temporary location used during construction/processing of savesets larger than 4MB.

Architectural Considerations

The following section will highlight architectural considerations for EV10 Indexing.

Overview of Index Server Groups

A new feature in Enterprise Vault 10 is the ability to split out Indexing and Storage roles. Index Servers can be grouped together into logical containers called Index Server Groups. Index Server Groups can be used to share resources for more scalable indexing and searching.

A dedicated Indexing server (i.e. a server that does not have a local Storage service or agents) must be a member of an Index Server Group in order to host indexes. A combination EV server (i.e. one that has both Index and Storage services) can also be part of an Index Server Group; this will however defeat the point of Index Server Groups as the server's resources will not be dedicated to indexing and might reduce performance.

Important note: The Index Server Group membership of associated index servers and vault stores **cannot be changed** as soon as the first item is archived. Index Server Groups and vault stores association can be changed or removed, but once data is archived you cannot change components, whether index server, vault store or indexes, in the group.

When to Use Index Server Groups

Index Server Groups are a scalability feature and as such should be considered carefully. In most cases a cost/benefit analysis will help determine whether Index Server Groups should be included in the design; weighing up the customer requirements for search performance and archiving throughput against the added complexity and cost of additional hardware and Windows server licenses.

In environments where frequent eDiscovery searches are performed on large index volumes such as journal or file system archives, and the speed of which results are returned is important, the use of Index Server

Groups will likely be of benefit. Index Server Groups provide the option to scale out index volumes, and use specialized fast storage and provide fast results when it matters.

The impact of increasing the number of servers in the Index Server Group does not improve archiving or expiry performance significantly, but will have a positive impact on Discovery Accelerator search performance.

Mailbox archiving and SharePoint archiving are unlikely to benefit from using Index Server Groups. This means that in the majority of cases we anticipate that Index Server Groups will not be required.

Splitting out Indexing Services from Storage Services

By adding additional hardware you can split storage services and indexing services between servers, freeing up resources that would have been consumed by other processes. This can be particularly beneficial where an EV server is already running at capacity.

Consider an environment where a single Exchange server is hosting 8,000 mailboxes, and archiving from a single EV server is the only option because one Exchange servers hosts all the mailboxes, or you are simply upgrading an EV9 server already running at capacity. In this scenario the EV server is likely to benefit from having indexing offloaded to a different server, as more resources will be available for storage processes by removing the CPU and memory overhead associated with indexing; this could result in archiving and indexing throughput increase by up to 30%.

There is however a balance in this scenario, where if the EV server is not running at capacity and there are sufficient available resources for indexing and storage on the single server, the benefit of offloading indexing will be offset by the fact that it is now remote to the storage server, and very little or no performance improvement will be seen as previously local processes are now remote.

Note that end user operations, such as Web Search, Vault Cache/Virtual Vault and Archive Explorer, will use the Storage server as a proxy for any operations requiring access to the index, so it is unlikely that those operations will benefit from offloading indexing to an Index Server Group. The one exception is in cases where an end user has access to many archives and/or many index volumes. The typical case is file system archiving – a user might have access to 50 index volumes and if these are distributed in an Index Server Group then it is likely that better performance will be achieved.

Scaling indexing out from storage requires good network connectivity; minimum 1Gbp/s network connectivity with very low latency is recommended. Do not attempt to scale indexing from storage across a slow link. For example, do not associate a vault with a centralized Index Server Group across a WAN.

Index Rollover

Index volumes, of which there can be more than one per archive (usually only for large archives such as Journaling and file system archiving) can be hosted by different Index Servers within the same Index Server Group. If using Discovery Accelerator, this can be an advantage as search requests can now automatically be

spread over more than one EV Index server. By default Discovery Accelerator will open 10 search threads to a single EV server, so there is a clear advantage in having more than one server hosting the index volumes for large archives as Discovery Accelerator will be able to create multiple connections of 10 threads per server.

When an index reaches capacity and requires a new volume (a process known as rollover), EV will attempt to distribute the index volumes evenly over all logical index locations within the group when it picks the location for the new index. A round robin algorithm is used to determine the new location for archives. Note that for mailbox archives requiring rollover the existing location will be favored. Because of the different use cases for searching, and the different methods used for rollover, it is not recommended to mix mailbox archives with other types of archives within an Index Server Group.

Associating a Vault Store with an Index Server Group

To control which archives are indexed by a particular index server, Vault Stores are associated with Index Server Groups. A vault store can only be associated with one Index Server Group, or if left ungrouped it will use the local indexing service. Note that if no local index service exists, and if the vault store is not associated with an Index Server Group, no data will be archived.

Ungrouped servers (servers residing in the Ungrouped Servers container in the Vault Administration console) will work in the same way as EV9 servers did, with no shared resources. As mentioned earlier Index Server Groups are intended as a scale-out feature, and should only be used when specifically required.

Making Changes to Index Server Group Configuration

As mentioned earlier in this document once the first item is archived, you cannot remove a server from an Index Server Group, or change the association of a vault store within an Index Server Group. Note that you can still change the index location within a group using the Change Location utility.

If you need to remove indexes from an unwanted group, a new Vault Store not associated with an Index Server Group can be created, and Move Archive can be used to move the archives from the Vault Store associated with the unwanted Index Server Group. This process will take some times to complete as you are effectively re-archiving the data for that archive.

Closed Index Locations

Previous versions of Enterprise Vault would allow an index location to be closed. However, this simply meant that no new Index Volumes would be created in the closed index location, but additional index data could still be added to existing Index Volumes.

When closing a 64-bit Index location, the location is now actually closed and no further data written to it. This concept is very similar to a closed Vault Store partition. No new index volumes or indexed data will be added to a closed index location. Daily backups of closed index locations will not be necessary as no new data will

be added. However, changes (such as deleted content or metadata updates) can still be made to a closed index location. Thus, it is recommended that weekly or monthly backups of closed index locations be performed.

Example Design Scenarios

This section will detail two common design scenarios and considerations for each scenario where the requirements change, and how those requirements will change the index design. Note that these are not design recommendations as such. EV Designs should be based on the number of messages, documents or files that need to be archived, the time available to do the archiving in and the size and type of the content being archived. Other factors that should be considered are the server environment (virtual or physical), client technology deployed (Virtual Vault/Manual Archiving), storage growth, eDiscovery requirements and classification.

Scenario 1A – Entry Level Exchange Mailbox and Journaling Environment

As an introduction to this section of the document, the following base-level scenario is used to describe a typical Exchange 2010 environment with EV Mailbox and Journal archiving.
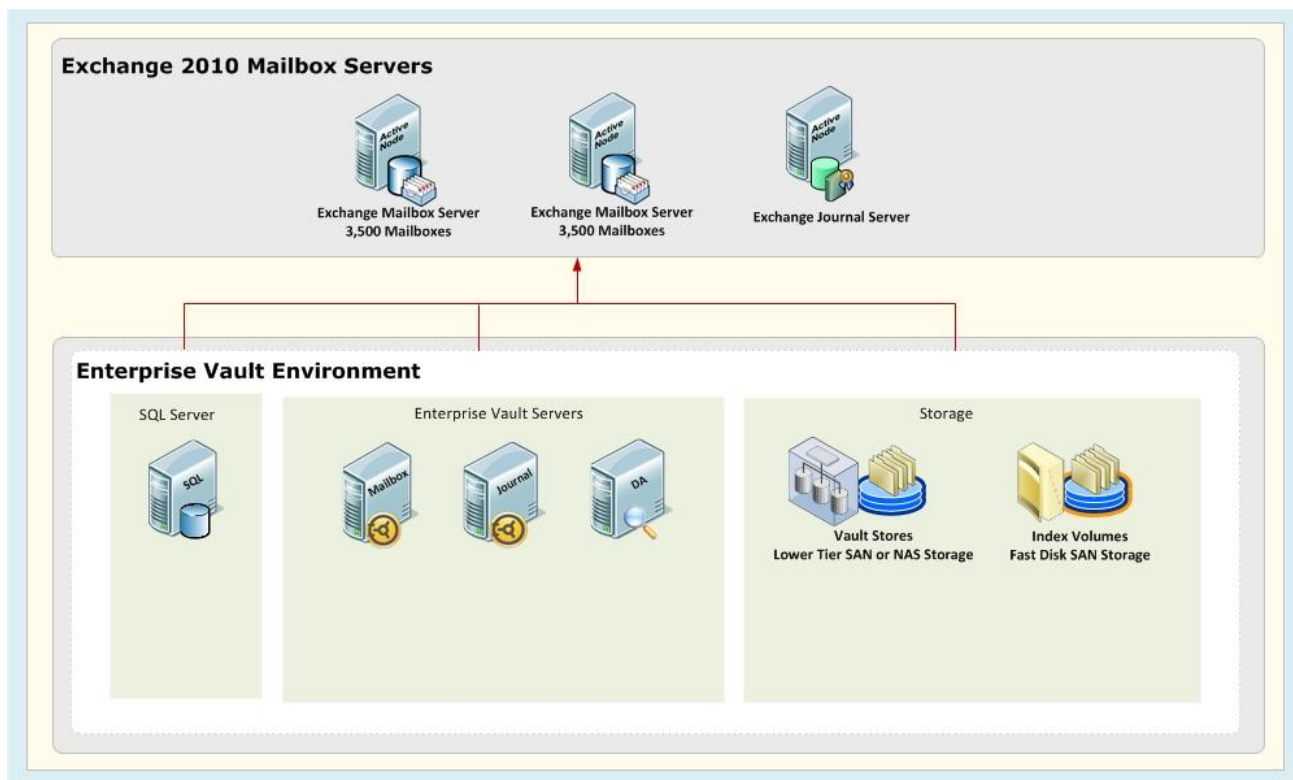
Environment overview:

- Exchange 2010:
  - 2 x Mailbox Servers, 3,500 Mailboxes per Exchange Server, 1 x Dedicated Exchange Server for Journal Mailbox[4]
- Enterprise Vault 10:
  - 1 x Mailbox Archiving Server, 1 x Journal Archiving Server, 1 Discovery Accelerator Server, SQL Server
  - Both Mailbox and Journal Archiving Server are running local Storage and Index services
- General Configuration:
  - 1 x Mailbox Vault Store, 1 x Journal Vault Store
- Storage:
  - Lower Tier SAN/NAS for Vault Stores, SAN Storage for Mailbox and Journal Index

In the next scenario we will look at one possible use of Index Server Groups.

---

[4] In similar environments the Journal mailbox(es) are hosted on the Exchange mailbox servers, however most customers prefer to host the journal mailbox on a separate server as this places less strain on the dedicated mailbox servers.

Scenario 1B – EV Server at Capacity, Splitting out Indexing from Storage

The following environment is similar to the first scenario, but with considerably more mailboxes hosted on each Exchange Mailbox server.
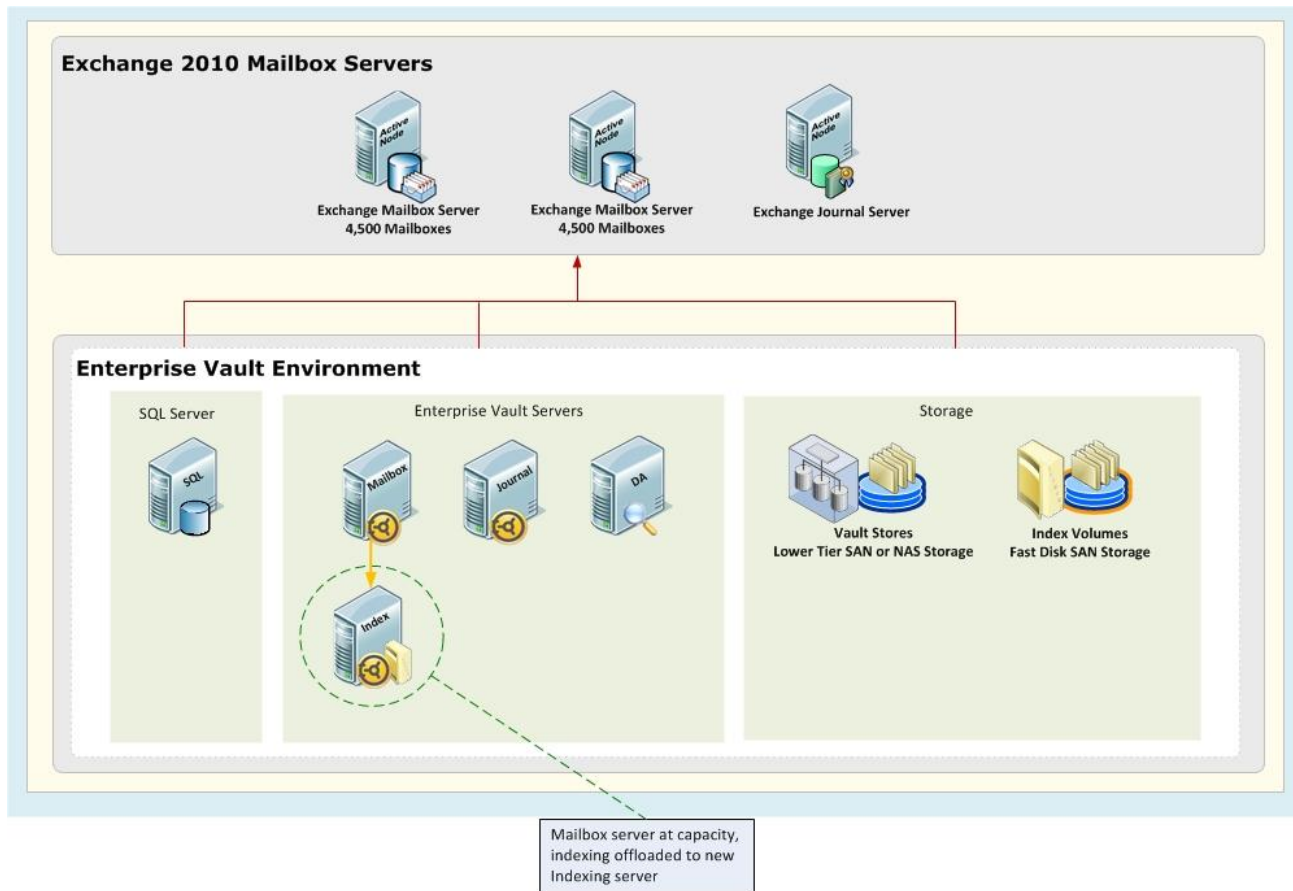
In environments where sudden growth has taken place within the company (such as mergers or acquisitions), or where an Exchange server consolidation took place during a migration, it is possible that the EV server is running at capacity and potentially not keeping up with archiving the daily change (and subsequently mailboxes are not being kept at a reasonable size).

In this scenario it is possible to add another EV server and use Move Archive to balance the archives across the servers (in addition to moving the associated mailbox archiving task so that a 1:1 Exchange/EV Server relationship exists). However, moving 4,000 archives is likely to take a long time.

In EV10 you now have the option to offload indexing to a new server by creating a new Index Server Group, adding a new EV10 Indexing server to the group, and moving the 64-bit Indexes from the Mailbox Archiving server to the new server. This will free up resources on the Mailbox storage server by removing the CPU and memory overhead associated with indexing, and could result in a throughput increase by up to 30% - enabling the EV server to keep up with the archiving load.

As discussed in an earlier section in this document there is a balance in this scenario, where if the EV server is not running at capacity, and there are sufficient available resources for indexing and storage on the single

server, the benefit of offloading indexing will be offset as it is now remote to the storage server, and very little or no performance improvement will be seen.

This scenario is more likely to be found in an upgraded environment, for a new design it would be more suitable to have a 1:1 Exchange to EV server, with 4,000 archives hosted on each EV server using a combined Storage/Index role.

Scenario 2A – Large Exchange Journal and Mailbox Archiving

In this new scenario we will look at a typical Exchange archiving environment, with a 1:1 Exchange Mailbox Server to EV Mailbox Archiving server relationship. The customer has very basic eDiscovery requirements and do not require regular large searches and instant access to results.
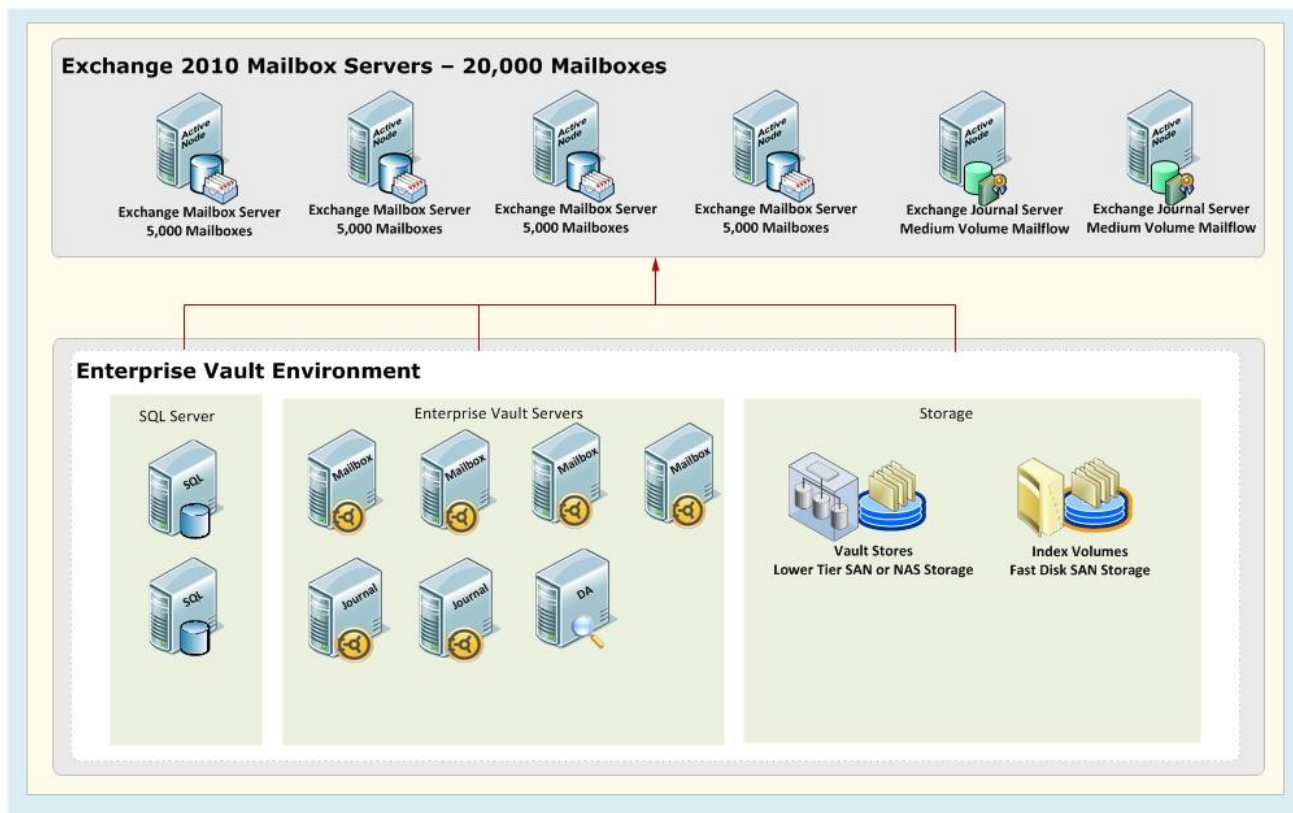

Environment details:

- Exchange 2010:
  - o 4 x Mailbox Servers, 5,000 Mailboxes per Exchange Server, 2 x Dedicated Exchange Servers for Journal Mailbox,
  - o Medium Journal mail volume
- Enterprise Vault 10:

- o 4 x Mailbox Archiving Servers, 2 x Journal Archiving Servers, 1 x Discovery Accelerator Server, 2 x SQL Servers
  - o Light eDiscovery requirements
- General Configuration:
  - o 1 x Mailbox Vault Store, 1 x Journal Vault Store
- Storage:
  - o Lower Tier SAN/NAS for Vault Stores, SAN Storage for Mailbox and Journal Index

In the next scenario we will look at how this design will change when faced with different end user requirements in a similar environment.

Scenario 2B – Large Journal Volume, Active eDiscovery Requirement
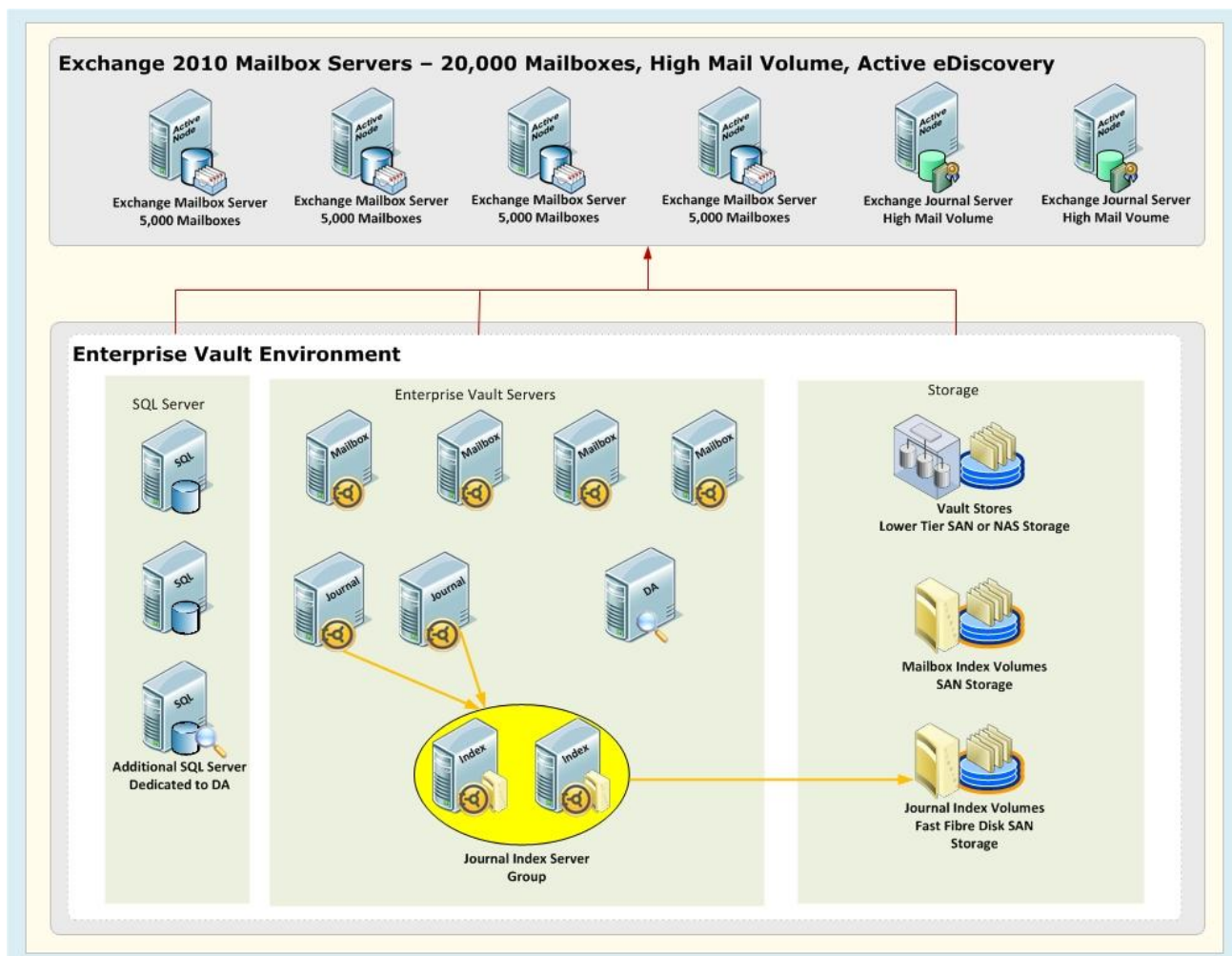
In this scenario the email administrator indicated that they have unusually high volume of emails passing through their Exchange servers, and the statistics they provided showed that they send and receive on average more than 80 messages per user per day. The administrator also indicated that they have more than

800 eDiscovery requests per year, and have 5 full time legal staff requiring access to the discovery search application. Rapid response to eDiscovery requests is critical to their business.

A key design consideration is how to design the environment so that it's able to meet the company's expectation for rapid response to eDiscovery requests.

A dedicated Index Server Group for Journaling will enable the company to scale out (by adding more Index servers) at a later date, and it will allow the administrator to be specific in the hardware design to be able to scale up (adding more CPU and memory). This requirement may dictate that Index servers are physical servers, and all other EV servers are virtualized in order to reduce infrastructure costs.

To further enhance the speed at which eDiscovery searches are performed a dedicated high performance Fiber Channel SAN disk array can be allocated to the Journal index locations.

A dedicated SQL server is added to the environment to host the Discovery Accelerator customer and analytics database to reduce the impact any eDiscovery searches will have on the environment.

Upgrading from Enterprise Vault 9

Planning for the Upgrade

Consider the following points and take action before you upgrade to EV10:

- Ensure your server is appropriately specified to meet the minimum or recommended CPU and Memory requirements
- Confirm the operating system is at minimum Windows 2008 R2 x64 – use the server settings migration wizard to migrate to a new server if required
- Medium level index volumes will require more space when the indexes are upgraded to 64-bit as they will map to full index (8% to 13%), ensure the index location is sized accordingly
- Brief level index volumes will also increase from 3% to 4% in most cases, confirm you have enough available disk space for the index location. Brief indexes for file archives will increase from 1% to 2%
- If using new Index locations, ensure the Anti-Virus exclusions (both real-time and scheduled) are added to avoid scanning those areas and cause open file locks. If adding new server cache locations ensure those locations are excluded from scanning
- Ensure your storage areas can handle the increase in IOPS requirements for the Index location and Server Cache areas

What happens immediately after the upgrade

It is very important to note that once the first item is indexed, the Index Server Group and vault store association is irreversible and cannot be undone easily.

The following processes will take place after the upgrade:

- Servers are put in the Ungrouped Index servers container (providing the same local indexing functionality as EV9)

- Existing 32-bit indexes are "frozen" with no new items added. Deletions and metadata modifications, such as movement of item in Virtual Vault, are still possible on 32-bit indexes.

- During the first archive run new empty 64-bit Indexes are created in the same index location - providing the existing location is still open, and there is a local index service usable by the storage service. If using a vault store with mailbox archives with an Index Server Group (and the Index Server Group is not co-existing with a storage service) then a round-robin mechanism will be used to allocate the mailbox index volumes

- If using Index Server Groups, journal indexes will be automatically allocated to available locations using a round robin algorithm

- Newly archived items go into 64-bit Index

- Searches are federated across 32-bit and 64-bit index

Note that the first mailbox archiving run using EV10 will be slow compared to subsequent archiving runs (both for a new installation and upgrades from V9), this is due to the new index volumes being created. A new volume will be created when the first item is indexed for that user.

Changes affecting End Users

Figure 1 shows search results from Enterprise Vault 9.0 using the default Attachment Granularity schema.

Figure 1 - EV9 search results using attachment granularity schema

Notice how the attachments are shown as separate search results from the top level document which is the message containing the attachments.

The 64-bit indexing engine no longer supports the Attachment Granularity schema and instead only supports the Item Granularity schema. The Item Granularity schema was configured in a large number of EV9 and earlier environments to improve search performance[5], specifically in Accelerator environments. The attachments are embedded with the parent message in the index and search results therefore only return top level items.

A further change in the end user interface is the sort order of search results. In Enterprise Vault 9 the default search order is relevance, in Enterprise Vault 10 the default sort order has been changed to "sent/received, newest first".

Upgrading 32-bit Indexes to 64-bit

For customers upgrading from EV9, the best search performance will be obtained if they upgrade existing indexes to 64-bit. There is however no reason to immediately perform the upgrades, searches are transparently federated across both 32-bit and 64-bit indexes so it recommended that initially only a small number (between 5 and 10) of indexes are upgraded and monitored. Once the administrator is comfortable with the process and an indication of upgrade performance is obtained a larger number of archives can be scheduled for upgrading. During the index upgrade process end-users and Accelerators can continue to search the old 32-bit index volumes until the upgrade is complete at which point it will be switched over to the 64-bit volume, and 32-bit volume deleted.

It is up to the administrator to decide which indexes they want to upgrade, and when the upgrades should take place. The default schedule is to run index upgrades during the day. Index upgrades can be performed throughout the night if required, including during archive runs.

Index upgrades cannot be performed while the Index service is in backup mode, therefore check the schedule of backups and when the backup mode PowerShell scripts are set to run.

After the upgrade the 32-bit index will automatically be deleted, unless errors were reported. The status console will clearly tell you if there was a problem upgrading the index, and you can investigate the logs and try again.

The upgrade does not require the 32-bit index to be functional as it is performing a rebuild index from storage. Slower storage (particularly partitions migrated to lower tier storage such as tape devices) will take considerably longer to rebuild. If the upgrade fails then it is most likely due to issues with the original data in storage. A failed upgrade will generate a report, and based on the content of the report you can either try and repair the storage or ignore the error and allow it to complete.

---

[5] Since EV8 SP3 the item granularity schema was also the default for Journal archives

Performance testing showed Exchange items rebuild at 366,000 per hour on EV10 recommended hardware, compared to 108,000 per hour on EV9.

Index Upgrade process

The Index Upgrade task enables index volumes to be upgraded from 32-bit to 64-bit. The 32-bit volumes remain searchable whilst the new volumes are being rebuilt.

The behavior of upgrade depends on the type of archive:

- Journal; 64-bit volumes created after the upgrade will remain as they are. Only 32-bit volumes are upgraded to 64-bit volumes
- Other archive types; In order to try and keep the number of index volumes small, existing 64-bit volumes will be merged to enable fewer volumes after the upgrade.

Index Upgrade Scenario 1: Mailbox archive, end-user search

For upgraded EV installations, the user will likely have had a single (32-bit) index volume before upgrade. After upgrade the user will have an additional 64-bit index volume, to be used for all newly indexed data. Searches from end-user applications (e.g. Outlook Integrated Web Search) will automatically federate across both volumes.

- For brand new EV installations, the user will have a single 64-bit volume
- On upgraded EV installations, the customer may decide to upgrade 32-bit indexes to 64-bit. When this happens for mailbox archives, the data is all consolidated into a single 64-bit volume where possible.

Index Upgrade Scenario 2: Journal archive, Accelerators search

For upgraded EV installations, the last 32-bit volume will be 'closed' and a new 64-bit volume created for any newly indexed data. Compliance and Discovery Accelerator will pick up this new 64-bit volume on its next archive synchronization, and search it in the same way as it searches 32-bit volumes. Both Discovery and Compliance Accelerator will perform its own federation, so from an EV point-of-view each search targets only a single volume, either a 32-bit or a 64-bit one.

On upgraded EV installations, the customer may decide to upgrade 32-bit indexes to 64-bit. When this happens for journal archives, the existing 32-bit indexes are rebuilt into 64-bit indexes, but the latest (64-bit) volume is left untouched.

Other Deployment Recommendations

Index Locations

The following general recommendations apply to index locations:

- If indexes are stored on NetApp devices, and possibly other NAS systems, opportunistic locking must be turned off for volumes that contain indexes
- It is best practice that a minimum of 8 index locations are created per index server. This is created automatically by default
- Given indexes can be distributed across an Index Server Group, this means that every index server in the group can then demand data from a single storage server, potentially overloading the server. It is recommended that Index Server Groups have no more than 15 members for Journal archives
- Disable Windows file indexing on the drives that contain Enterprise Vault indexes.

Disable Anti-Virus for Index Locations

Ensure that there is no Anti-Virus scanning the index locations, real-time or scheduled. Failing to exclude the indexes from scanning can cause index corruption and will result in reduced server performance. A new requirement in EV10 is to exclude the Indexing meta-data folder, default location "<install path>\EVIndexing\data\indexmetadata".

Pagefile Size

The Pagefile should conform to Microsoft's recommendation of 1.5 times the amount of memory. It is recommended that the page file is placed on a different partition and different physical hard disk drive from the system partition so that Windows can handle multiple I/O requests more efficiently.

Additionally, if you set the "Initial size" of the Pagefile to the same as the "Maximum size" it will prevent unnecessary fragmentation file. An un-fragmented paging file leads to faster virtual memory access.

Index Server Shutdown

When using Index Server Groups in larger complex environments it can be more difficult to understand the impact of shutting down a particular index server for patching/maintenance purposes. The following SQL query can be used to determine which index volumes will be affected by shutting down a particular server:

```
SELECT DISTINCT ArchiveName FROM view_IndexVolumeServer WHERE
ComputerNameAlternate = '<Index server name>'
```

Index Volume Fragmentation

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 need to be managed on any index

storage device. Either an automated background file defragmentation product or scheduled device defragmentation must be employed.

Index Backup

Detailed recommendations on how to back up the index volumes can be found in the Enterprise Vault Best Practice Guide for Backups (July 2011).

Tuning

Federated Searching

When searches are performed (using Discovery Accelerator, Browser Search, Outlook Integrated Search, etc.), all Index Volumes for an archive (both 32-bit and 64-bit) will be searched automatically. Discovery Accelerator will always search all Index Volumes separately and combine the results. End-user search applications (such as Browser Search) will only search a maximum of five index locations concurrently by default. This setting can be changed by modifying the FederatedSearchMaxVolSets value in the webapp.ini file. As with all performance tuning care should be taken when changing default values – changes should be made in a controlled fashion and any impact measured against baseline performance testing done beforehand.

Adding more CPU vs. Adding more Memory

Scaling up an index server can help improve indexing performance; generally more performance will be obtained by adding more CPU as opposed to more memory. If disk I/O is proving to be a bottleneck adding more memory will help improve performance.

Performance Counters

A number of Windows performance counters have been added under an "Enterprise Vault Indexing Volumes Processing" category. These counters will capture the ingestion state of affairs since the last Indexing service restart.

- **Additions Items**: The number of items successfully indexed to the local indexing engine
- **Processed Items**: The number of items processed for indexing. This includes items that have been successfully indexed as well as the failed items and items that have been indexed but are not yet confirmed
- **Deleted Items**: The number of items deleted from the local indexing engine
- **Updated Items**: The number of items updated in the local indexing engine

The following table described counters tracking activity within the agents.

Performance Counter	Description
Item fetched	Items retrieved from storage
Item submitted	Items submitted to the index engine but not yet confirmed as indexed
Item acknowledged	Items confirmed by indexing engine as indexed
Item audited (failed)	Item recorded as failed
Item audited (successful)	Items recorded as successful
Item persisted (pending)	Items recorded as pending acknowledgement (because of shut down or re-queue). These will be reloaded and acknowledged later
Item reloaded (pending)	Items previously persisted as pending acknowledgement that have been reloaded for acknowledgement
Item in the pipeline	Total items currently in the index pipeline
Item in the pipeline	Items that have re-fetched
Item missing from the pipeline	Items that have been re-acknowledged
Item queue count	Total items currently queued up waiting to be traversed ready to send
Item queue size (KB)	Total memory used by items queued up waiting to be traversed

Table 4 - Counters tracking activity within the Index engine

See additional performance counters available in Appendix C.

APPENDIX A – How an EV Search Works

The following scenario will describe the End-user search process in detail.

1. User opens Search.asp through Outlook or Internet Browser, enters search criteria and clicks on the Search button

2. Search.asp request is processed by the W3WP.exe process on the EV web server.

3. Search.asp loads IndexClient.dll (into the W3WP process), using the DCOM and submits the search using the IndexClient API

4. IndexClient loads the index volume information for the archive from the IndexVolume table in the EV directory.

5. For each index volume in the archive, IndexClient contacts the relevant index server and submits a search request in parallel. This is only the case for federated searches - other applications would have to search each index volume separately.

6. For 32-bit volumes on the server processing the request, IndexClient uses DCOM to connect to IndexBroker, which connects launches an IndexServer process and returns a reference to the IndexServer process to IndexClient.   IndexClient then executes the search against IndexServer using DCOM

7. For 64-bit volumes on the server processing the request, IndexClient uses HTTP to connect directly to EVIndexQueryServer and execute the search

8. Server-side indexing performs the search against the indexing engine:

9. For 32-bit searches, IndexServer loads the 32-bit engine, performs the search and returns the results to IndexClient

10. For 64-bit searches, the query goes via the collection-broker.exe to the query-service.exe process, which in turn loads the right collection by using collection- service

11. IndexClient gathers together the results from the one or more searches, merges them if necessary, then returns them to the client application (in this case search.asp)

APPENDIX B – Index Administration Tasks

This section will outline some of the new Administration features available with the 64-bit index engine, and describe the tasks available on the "Manage Indexes" wizard in more detail.
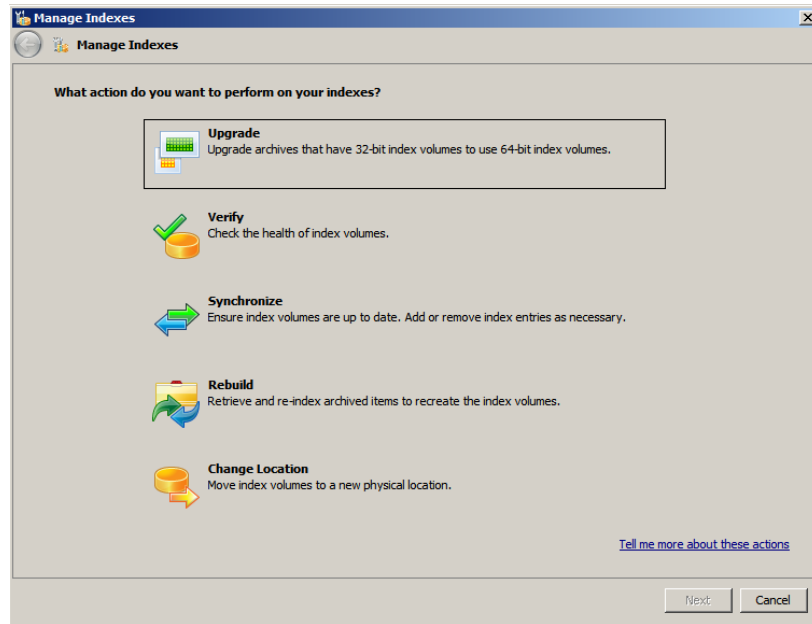
A new 'Indexing Administration Task' is responsible for starting and stopping indexing sub-tasks, controlled from the Vault Administration Console (VAC).

Upgrade

Upgrade allows index volumes to be upgraded from 32-bit to 64-bit. The 32-bit volumes remain searchable whilst the new volumes are being rebuilt.

The behaviour of upgrade depends on the type of archive:

- Journal; Any existing 64 bit volumes are skipped as part of the upgrade. Only 32-bit volumes are upgraded to 64 bit volumes

- Other archive types; In order to try and keep the number of index volumes small, existing 64 volumes will be rebuilt too enabling fewer volumes after the upgrade.

An index is built seamlessly by maintaining the original volume(s) in a Read-Only state whilst a new volume is being built. Only when the new volume(s) has been built successfully will the old one(s) be deleted and replaced by the newer volume. The whole procedure is largely transparent to search applications, except for the potential need to refresh cached index volume information once the upgrade is complete.

The following steps describe the process of upgrading a 32-bit index:

- Validate sequence numbers. If rebuilding an archive then the sequence numbers of all volumes are checked to ensure they are in a continuous range

- If rebuilding an entire archive and the last volume is empty, then it will be removed. This is done in order to end up with the fewest volumes possible

- Set the live index volume as read only

- Create a new hidden volume (one or more depending on the size of the original volume). This follows the standard Round Robin approach to first create the database entry and then allocate a physical Index server to process the Index volume

- o When upgrading, existing 64-bit journal archive archives will be ignored. For other archive types the 64-bit volumes will be included in the upgrade.
- Allocate a root path for the volume (this may be on a different server if using indexing groups)
  - o If on a different server, ownership of the subtask is transferred to the new server
- Ingest items into the volume
- At the end of the volume:
  - o If all items were indexed successfully
    - ▪ The old volume is deleted and the new volume made visible
    - ▪ If all of the original volume could not fit in the new volume, then the subtask moves back to the create hidden index volume step and creates another to hold more of the volume
  - o If there were errors with some of the items in the volume, then the option chosen in the wizard will apply:
    - ▪ If ignore errors were selected, EV will continue and switch over to the new volume, or create another hidden volume for more of the data

If ignore errors is not set, the subtask moves to a Waiting state and will remain in that state until the Administrator user chooses how to respond to the failure.

Failures of the rebuild and upgrade subtask can be in different forms:

- Individual item errors
- Frenzy of errors. If the subtask encounters a number of consecutive items that have failed then the subtask will subtask will be failed and marked with a frenzy error. The user can retry the subtask when it is in this state, the task will continue from where it left off and retrying these items again
- Non Critical errors – These will cause the subtask to fail and be re-queued for processed a number of times before the subtask is failed. Non critical errors would be errors that we would consider intermittent. Again, a retry is available
- Critical errors – These errors cause the subtask to be immediately failed and are reserved for errors that are unlikely to be fixed with a retry (for example if the storage is not available). A retry will be available to allow the subtask to be retried but the administrator would normally have to change something in order for it to work.

Verify Operation

Verify replaces and enhances the "IndexCheck" functionality found in previous versions of Enterprise Vault. The tool is able to check the integrity of an index and will report on errors and inconsistencies.

Verifying an index volume involves the following steps:

- Checks that the subtask is running on the correct computer
- Checking the failed volume reason
- If the volume is found to be offline, then the subtask will be set to "Will Retry" and will be retried again in 30 minutes.
- Set volume read-only. This is done to block additions, updates and deletions on the volume whilst verifying the contents
- Confirm that the physical location of the volume to ensure it exists, if not found an error would be reported and the subtask will fail
- Checking collection exists

- Checks that the highest sequence number in the index is not higher than the highest sequence number in storage
    - This indicates that the storage database has been restored and the index is ahead. This will stop the task and report to the administrator that the EVSVR tool should be run to correct the problem
- Report if the volume contains mixed schema (this information comes from the database and not by looking at the volume)
- If the "Complete" option were selected, missing items will be checked and reported on
- The volume is set to read-write
- Create synchronize subtask. If running a complete verify and some missing or extra items have been found, a synchronize subtask will be automatically created to attempt the repair. This will only be done if the option was chosen in the wizard.

Change Location

This subtask is used to move the index volumes for an archive to a new location. In previous versions of Enterprise Vault Administrators used to be able to copy the index files over and update the database.

With the new 64 bit volumes, it's more complicated as metadata collections must exist for the index volume so administrators should not manually copy data over and update the directory database to perform a move.

NOTE: Users must still copy the index files over, but "Change Location" allows it to be done in a controlled way.

Rebuild

The biggest enhancement in Rebuild is that rebuilds are now performed in the background, allowing the existing index volume to be searched whilst the new volumes are being rebuilt. In previous versions, the old index volumes were deleted at the start of the rebuild. Rebuilding a 32-bit index volume will rebuild the volume as a new 64 bit volume.

The behaviour of rebuild depends on the type of archive:

- Journal; if rebuilding the entire archive or the live index volume a new live index volume is created before the existing archives are rebuilt. This is to avoid the situation where the rebuild never catches up as new data is archived and indexed
- Other archive types; No new live index volume is created; existing volume(s) are rebuilt as 64 bit index volumes

The rebuild involves the following steps:

- Check Sequence numbers
- Create a new hidden volume (we may need more than one depending on the size of the original volume). This follows the standard Round Robin approach to first create the database entry and then allocate a physical Index server to process the Index volume
- Set the source volume read only (if not already set) to avoid any additions, updates or deletions
- Allocate a root path for the volume (this may be on a different server if using indexing groups)
    - If on a different server, ownership of the subtask is transferred to the new server
- Ingest items into the volume
- At the end of the volume:
    - If all items were indexed successfully the old volume is deleted and the new volume made visible
    - If all of the original volume could not fit in the new volume, then the subtask moves back to the create hidden index volume step and creates another to hold more of the volume

o If there were errors with some of the items in the volume, then the option chosen in the wizard will apply: If ignore errors, EV will continue and switch over to the new volume, or create another hidden volume for more of the data. If ignore errors is not set, the subtask moves to a Waiting state and will remain in that state until the user chooses how to respond.

Synchronize

The Synchronize subtask replaces the "Repair and Upgrade" functionality. The task attempts to repair poison pilled additions and deletions, followed up an internal sync to get the volume up to date.

The Repair and/or Update functionality is now processed as a 'Synchronize' subtask.

Synchronizing a 64 bit volume involves the following steps:

- Checks that the subtask is running on the correct computer
- Analyze the failed volume reason if volume is failed
  - If error indicates state file is missing or corrupt then the repair will check the state file again to see if it exists and is valid before failing the subtask
  - If the failed volume reason is due to a sequence number, then it'll be checked again and the synchronize will be failed if there is still a mismatch
  - If the failed volume reason is a metadata sync error from index admin startup, then the subtask will fail. The Index Admin server will attempt to fix this at next startup
  - For other reasons, the error code will be cleared and the repair will be attempted
- Verify that the sequence number in the index is not higher than that in storage. If so volume will be marked as failed with appropriate reason
- Attempt to repair poison pilled items that are missing from the index
- Attempt to repair poison pilled extra items that are in the index
- Trigger an internal sync to process any non-poison pilled additions/updates/deletions and ensure volume is up to date

Synchronizing a 32-bit volume involves the following steps:

- If the volume is marked as failed and the failure reason is not deemed to be a temporary reason, then the information is written to the report and the subtask failed
- Verify that the sequence number in the index is not higher than that in storage. If so volume will be marked as failed with appropriate reason
- Process any outstanding updates on the index volume
- Process any outstanding additions on the index volume
- Attempt to repair any failed (poison pilled) additions
- Process any outstanding deletions on the index volume

The existing retry of failed index volumes every 6 (Extended setting in VAC) hours remains.

If too many errors or a critical error occurs, this will result in the task being marked as failed. The report file will indicate how many items have been added and deleted from the volume and will also report on how many are still failed.

APPENDIX C – Additional Performance Counters

Counters tracking movements of work items through the Index queue system:

- Average agent processing time (seconds)
- Average delete index volumes queue time (seconds)
- Average sub task queue time (seconds)
- Average synchronization queue time (seconds)
- Average task queue time (seconds)
- Average work item queue time (seconds)

Counters tracking the size of the Index queues:

- Delete index volumes queue
- Number of indexing agents
- Sub task queue count
- Synchronization queue count
- Task queue count
- Work item queue count

Index Query Server Counters

- Average query time (seconds)
- Average search engine time (seconds)
- Failed queries
- Failed queries due to search engine out of resources
- Queries
- Queries in progress
- Queries per second
- Queued queries
- Results
- Results per second
- Search threads

About Symantec:

Symantec is a global leader in providing storage, security and systems management solutions to help consumers and organizations secure and manage their information-driven world.

Our software and services protect against more risks at more points, more completely and efficiently, enabling confidence wherever information is used or stored.

For specific country offices and contact numbers, please visit our Web site: **www.symantec.com**

Symantec Corporation
World Headquarters
350 Ellis Street
Mountain View, CA 94043 USA
+1 (650) 527 8000
+1 (800) 721 3934