

Symantec Discovery Accelerator™

Best Practices for Implementation

10.0

Last updated: May 1, 2013



Symantec Discovery Accelerator™: Best Practices for Implementation

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

Last updated: May 1, 2013.

Legal Notice

Copyright © 2013 Symantec Corporation. All rights reserved.

Symantec, the Symantec Logo, the Checkmark Logo, Enterprise Vault, Compliance Accelerator, and Discovery Accelerator are trademarks or registered trademarks of Symantec Corporation or its affiliates in the U.S. and other countries. Other names may be trademarks of their respective owners.

This Symantec product may contain third party software for which Symantec is required to provide attribution to the third party ("Third Party Programs"). Some of the Third Party Programs are available under open source or free software licenses. The License Agreement accompanying the Software does not alter any rights or obligations you may have under those open source or free software licenses. Please see the *Third Party Software* file accompanying this Symantec product for more information on the Third Party Programs.

The product described in this document is distributed under licenses restricting its use, copying, distribution, and decompilation/reverse engineering. No part of this document may be reproduced in any form by any means without prior written authorization of Symantec Corporation and its licensors, if any.

THE 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.

The Licensed Software and Documentation are deemed to be commercial computer software as defined in FAR 12.212 and subject to restricted rights as defined in FAR Section 52.227-19 "Commercial Computer Software - Restricted Rights" and DFARS 227.7202, "Rights in Commercial Computer Software or Commercial Computer Software Documentation", as applicable, and any successor regulations. Any use, modification, reproduction release, performance, display or disclosure of the Licensed Software and Documentation by the U.S. Government shall be solely in accordance with the terms of this Agreement.

Symantec Corporation
350 Ellis Street, Mountain View, CA 94043
www.symantec.com

Contents

Chapter 1	Introduction	7
	Purpose.....	7
	Customer profiles.....	8
	Small customer.....	8
	Medium customer	8
	Large customer.....	8
	General hardware considerations	9
	CPU	9
	Memory.....	9
	Storage.....	10
Chapter 2	Database platform	11
	Hardware requirements	11
	CPU considerations.....	11
	Memory considerations.....	12
	Network considerations	13
	Storage considerations.....	14
	Virtualized infrastructure	21
	Database tuning	21
	SQL permissions.....	22
	Deploying databases and the impact of model database	22
	The tempdb database.....	24
	Database maintenance	25
	Database upgrades	27
	Advanced database tuning	27
	Multiple SQL Server instances.....	27
	Distributing I/O with multiple database files	28
	Moving Discovery Accelerator tables or indexes into file groups	29
	Rolling over customer databases.....	29
	Co-locating multiple customer databases.....	30
	Monitoring the Discovery database server	30
	CPU and memory.....	32
	Disk.....	33
	SQL Server.....	33
	Useful queries	35

Chapter 3 Discovery Accelerator server..... 43

Hardware requirements	43
Server considerations	44
Storage considerations	45
Virtualized infrastructure.....	46
Custodian Manager tuning.....	47
Discovery Accelerator customers service tuning.....	47
Network sockets	48
Discovery Accelerator analytics service tuning	48
Retrieval thread tuning	49
Discovery Accelerator customer tasks tuning	49
Pre-fetch cache tuning.....	50
Search tuning	51
Export and production tuning	54
Architecture considerations	55
Using multiple customer databases.....	56
Using multiple Discovery installations.....	57
Monitoring Discovery Accelerator.....	58
CPU and memory	59
Disk	60

Chapter 4 Discovery Collector 61

Hardware requirements	61
Server considerations	63
Storage considerations	63
Discovery Collector tuning.....	63
Desktop collection	64
Architecture considerations	64
Monitoring Discovery Collector	64

Chapter 5 Enterprise Vault infrastructure..... 65

Enterprise Vault indexing service considerations	65
Server considerations	66
Storage considerations	67
32-bit and 64-bit index volumes.....	71
64-bit index volume tuning.....	72
32-bit index volume tuning.....	72
Index file locations	75
Index file fragmentation.....	76
System Tuning	76
Enterprise Vault storage service considerations.....	76

Server considerations.....	77
Storage considerations.....	78
Scaling Enterprise Vault for Discovery Accelerator.....	79
Scaling out.....	79
Rolling over journal vaults	80
Monitoring Enterprise Vault.....	80
CPU and memory.....	81
Disk.....	81

Chapter 6 End-user advice 83

Legal holds	83
Searching.....	83
Analytics.....	84
Reviewing	84
Export and production	84

Introduction

Purpose

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

Discovery Accelerator requires the proper consideration in the following areas:

- Discovery database server
- Discovery application server
- Discovery collector infrastructure
- Enterprise Vault infrastructure, including indexing and storage services

The purpose of this guide is to discuss the different aspects that you need to consider during sizing and recommend best practices for implementation.

The Discovery Accelerator business users and the team designing a solution need to discuss the Discovery Accelerator model, paying attention to the anticipated number of active cases, simultaneous users, potential volume of review, and any other litigation support applications with which data must be exchanged. Symantec publishes several white papers that explore the different options and use models.

This guide assumes that you are familiar with how to configure and administer Discovery Accelerator and associated products. You can obtain more detailed installation and configuration information from the Discovery Accelerator documentation. Symantec also publishes several white papers that explore specific details such as effective searching, security, and more.

Customer profiles

Small customer

- 100 mailbox users with both journal and mailbox archiving.
- Requires a single server for Enterprise Vault and Discovery Accelerator.
- Server storage needs to be local or directly attached device.
- One legal Discovery user (case administrator).
- 10–20 active cases per year.
- Low search and production load, and single online reviewer.
- Likely to have one or two cases in analytics at any time.

Medium customer

- 10,000+ mailbox users with both journal and mailbox archiving.
- Large volume of historical data to archive.
- Environment spread over multiple sites.
- Rapid and large or critical responses required.
- Legal Discovery team likely to have 4–10 members.
- 50–100 active cases per year.
- High search and production load, and 10 online reviewers.
- Likely to have up to 25 cases in analytics at any time.

Large customer

- 30,000+ mailbox users with both journal and mailbox archiving.
- Large volume of historical data to archive.
- Environment spread over multiple sites.
- Rapid and large or critical responses required in high pressure environment.
- Legal Discovery team likely to have 5+ full-time members.
- 1000+ active cases per year.
- Very high search and production load, and 20 – 30 online reviewers.
- Likely to have up to 50 cases in analytics at any time.

General hardware considerations

Apply these hardware considerations to all servers deployed as part of the Discovery Accelerator environment.

CPU

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

Hyper-threading technology is claimed to provide up to 30% improvement in performance. These processors contain two architectural states on a single processor core, making each physical processor act as two logical processors. However, the two logical processors must share the execution resources of the processor core, so performance gains may not be attained. The operating system, and potentially the application software, needs to support hyper-threading to avoid inefficient use of the logical processors. Windows server operating systems support hyper-threading.

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

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

Memory

A 64-bit platform should be considered for memory-intensive applications, if supported. Using a 64-bit platform can provide more efficient memory utilization, which can bring performance benefits.

Storage

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

RAID levels 5 and 6 are popular, cost-effective methods of achieving redundancy while maintaining striped disk read performance. However, writing incurs a cost of four to six physical operations per write. A poorly sized RAID-5/6 implementation can significantly reduce the performance of write-intensive activity. Correctly sizing a RAID-5/6 implementation to maintain write performance may become more costly than RAID-10. Therefore, in some cases discussed below, 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 the multiple storage locations and provide sufficient throughput. The controllers should also provide a battery-backed read and write cache to aid performance.

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

When you create a basic NTFS volume on the storage device, align the partition with the device track sectors to prevent unnecessary disk operations. (Dynamic volumes cannot be aligned at time of publication.) You can achieve this through the Windows 2003 and Windows 2008 command line tool diskpart. See the Windows Help or Microsoft TechNet for more information.

Database platform

The Discovery Accelerator database servers should be sized, tuned and maintained according to Microsoft best practice advice for SQL Server. This guide discusses some of those best practices from a Discovery Accelerator perspective and should be used in conjunction with Microsoft advice.

The Discovery Accelerator database performs the majority of data processing, and this tends to be very resource-intensive. The high level of activity will struggle to co-exist with any other database application, and, during Discovery Accelerator implementation, you must pay special attention to the database server and its configuration.

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

Hardware requirements

SQL Server benefits from multiple processors, particularly with the parallel activity generated by the Discovery Accelerator service. The flow of data within the Discovery Accelerator service, and the manipulation of data at the Discovery Accelerator database, can cause high memory utilization. Most Discovery Accelerator actions result in updates to the database, and the volume of manipulation means that the I/O subsystem has to handle a high level of activity.

CPU considerations

A Discovery Accelerator customer database generally uses four CPUs during Discovery search, review, and export activity, and this also depends on the database server I/O subsystem throughput. In addition, collecting analytics data for a single case can require the use of more CPUs.

Therefore, a SQL Server instance that is hosting a Discovery Accelerator customer database should have a minimum of four CPUs dedicated to the SQL Server instance – either four physical CPUs or similar combination of multi-core CPUs, but not based on hyper-threading. Medium and large customers may wish to consider eight CPUs.

Note: Hyper-threading is not recommended to improve the database performance due to potential performance problems when the database places a load on the memory. See the MSDN article "Be aware: To Hyper or not to Hyper" for further information.

You can host the Discovery Accelerator database on either 32-bit or 64-bit (x64 only) platforms. The 64-bit platform can provide performance benefits due to memory enhancements.

In most cases, the SQL Server instance should manage the CPU resources. Do not set the CPU affinity mask unless absolutely necessary, as this can significantly impact the performance. When you run multiple SQL Server instances, the most common reason for setting the CPU affinity mask is to prevent an instance being starved of resources (see "Multiple SQL Server instances" on page 27).

Memory considerations

The Discovery Accelerator stored procedures employ methods that are memory-intensive. In addition, the flow of data between many tables and their indexes causes a large number of pages to be required in memory at any time, consuming large volumes of memory. The Discovery Accelerator analytics feature can potentially insert large text documents into tables, which results in very high memory use and corresponding I/O activity.

The SQL Server instance that hosts the Discovery customer database requires enough memory to ensure that the data manipulation does not cause excessive paging to disk both at the Discovery database and tempdb, which will quickly degrade the performance. A small environment should have a minimum of 4 GB of RAM, whereas a medium or large environment should have at least 8 GB of RAM available to the SQL Server instance.

The use of analytics places particular pressure on the memory. If users enable multiple cases for analytics simultaneously, the collection of analytics data in each case applies very high pressure on memory. This is reflected in very high I/O activity, high CPU, and potentially a reduction in throughput. To reduce the I/O load and maintain the throughput, you can scale up the database server memory by 4 GB per anticipated concurrent analytics data collection. A 64-bit database server is recommended.

Using an x64-based 64-bit platform provides more efficient memory utilization and brings performance benefits. Install the appropriate edition of Windows Server and SQL Server to support the capacity of memory that you have installed.

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

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

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

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

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

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

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

Network considerations

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

It is also recommended that the TCP Chimney Offload, TCP/IP Offload Engine (TOE) and TCP Segmentation Offload (TSO) are disabled. See Symantec technical article [TECH55653](#).

Storage considerations

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

Each database requires the disks to be arranged for two different purposes: the database data files, and the log files. The data files require good random access or high number of IOPS, and therefore a striped array of many disks should be used (using hardware-based RAID rather than software-based RAID). The log files require good sequential write performance, so each log file should be placed on its own high speed array with good transfer rates.

To achieve redundancy on the sequential write-intensive disks (log), use a RAID-10 scheme with high speed, 15k rpm disks.

In the case of local or direct attached storage, use multiple controllers that support multiple channels to distribute the load and provide sufficient throughput. The controllers should provide a large capacity battery-backed read and write cache. A 512 MB controller cache is recommended for local or direct attached storage.

When you select storage, consider the advice in "Storage" on page 10, and create and align partitions as advised. See the TechNet article "SQL Server best practices". This article also recommends that you format both log and data partitions with 64 KB allocation unit sizes.

Arrange the database server storage to accommodate the different types of data. Typically, database servers should have the following partitions.

Recommended partitions for database servers

Partition	RAID array
System drive	RAID-1 array
Tempdb log partition	RAID-1 or 10 array
Tempdb data partition	RAID-10 striped array of several drives
Configuration database and Custodian Manager log file partition	RAID-1 or 10 array
Configuration database and Custodian Manager database data file partition	RAID-10 striped array of many drives
Each customer database log file partition	RAID-1 or 10 array
Each customer database data file partition	RAID-10 striped array of many drives

One or more analytics data file partitions, which can also serve as Full Text Index file locations	RAID-10 striped array of many drives
One or more analytics full-text index file partitions (optional)	RAID-10 striped array of many drives

Ensure that the partitions only contain the related database files so that they do not become fragmented on disk. If multiple database files are located on one partition, it is likely to require regular file defragmentation to maintain performance.

When the Discovery Accelerator customer database is created, alter the default data file size to a large value, perhaps representing the next year’s activity. This prevents file fragmentation and wasted I/O and waits while growing the files.

Consider the storage capacity of the Discovery Accelerator database server carefully.

Sizing the configuration database

The Discovery Accelerator configuration database stores details of the following:

- All registered customer databases
- The Custodian Manager database
- All customer configuration options
- Errors that the service has logged

The database is also used to manage analytics data collection tasks across all customers and cases. In general, the configuration database remains less than 10 MB. However, error conditions can quickly grow the related table. A good rule of thumb is to allow at least 100 MB for the configuration database data file.

Sizing the Custodian Manager database

The Discovery Accelerator Custodian Manager database stores details of the following:

- All custodians, including their attributes, email addresses, and group membership.
- Historical record of changes to custodian attributes, addresses, or groups over time.

The Custodian Manager synchronizes custodian details with the corporate directory infrastructure through Active Directory, Domino Directory, or LDAP queries.

Use the following rules of thumb to size the Custodian Manager database:

Base capacity (MB) = ((3.09e)+(0.23ea)+(2.52g)+(0.23ag)+(0.17en)+(0.05et))/1000

Yearly capacity (MB) = ((0.238ace)+(0.24acg)+(0.05m)+(0.05ect))/1000

Where:

- e Total employees in directory sources.
- a Average number of email addresses per employee/group.
- c Average number of address changes per employee per year.
- g Total number of groups.
- n Average group membership per employee.
- m Typical total number of group movements per year.
- t Average number of custom attributes (in addition to standard attributes).

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

When the Custodian Manager is synchronizing custodian details, the database typically requires 100 to 300 IOPS.

The following is an example medium scenario. 30,000 employees are managed within the corporate directory, each with three email addresses and two custom attributes. There are a total of 20,000 groups (based upon distribution lists), and each employee is on average a member of 10 groups. Across all groups there are typically 6,000 changes to group membership per year. The email addresses of users and groups tend to change on average once every 5 years (therefore, 0.2 times per year).

Base capacity (MB) = ((3.09e)+(0.23ea)+(2.52g)+(0.23ag)+(0.17en)+(0.05et))/1000

= (3.09*30,000)

+ (0.23*30,000*3)

+ (2.52*20,000)

+ (0.23*20,000*3)

+ (0.17*30,000*10)

+ (0.05*30,000*2)

/1000

=231.6 MB

$$\begin{aligned}
 \text{Yearly capacity (MB)} &= ((0.238\text{ace})+(0.24\text{acg})+(0.05\text{m})+(0.05\text{ect}))/1000 \\
 &= (0.238*3*0.2*30,000) \\
 &\quad +(0.24*3*0.2*20,000) \\
 &\quad +(0.05*6,000) \\
 &\quad +(0.05*30,000*0.2*2) \\
 &\quad /1000 \\
 &= 8.07 \text{ MB}
 \end{aligned}$$

Therefore, the Custodian Manager database needs a base of 231.6 MB and an additional 8.07 MB for the first year, plus 20% extra headroom. The total is 287.61 MB.

Sizing the customer database

Use the following rule of thumb to size the core Discovery customer database (excluding analytics tables):

$$\text{Capacity per year (MB)} = ((2.53\text{cis})+(0.44\text{acs})+(1.82\text{is})+(11.6\text{ce})+(0.222\text{cism}))/1000$$

Where:

c	Average cases per year
s	Average number of searches per case
i	Average number of items captured per search
a	Average number of archives searched per case
m	Average number of review marks per captured item
e	Average number of items exported or produced per case per year

Note: This calculation provides a high-level estimate only. It does not take into account the operating system, paging, and log file devices. It also does not include any additional capacity that may be required during product version upgrades. However, this calculation does allow for a small proportion of transient storage required for unaccepted searches.

On a server with 8 GB of RAM, a Discovery customer database data file typically requires 250 to 1,800 IOPS and log file around 500 to 1,200 IOPS. However, this varies depending upon concurrent activities and server specification.

In the following example, there is an average of 100 cases per year, with five searches per case normally searching 50 journal archives. Each search identifies 3,000 items on average (a total of 15,000 per case or 1.5 million items over all cases, per year). Each item has on average a single review mark, and 33% of the items are exported or produced (5,000 items per case).

Estimating the size of the analytics tables for each enabled case can become complicated when you take into account all the different characteristics of the source data: volume, size distribution, recipient volume and distribution, attachment volume, types and distribution, and number of unique conversations. The Discovery Accelerator source table tblIntDiscoveredItems contains some relevant details, but these are not sufficient to provide estimated sizes.

You must size the file group partitions to encapsulate multiple enabled cases of varied size with potentially varied characteristics. So, you must make a high-level estimate that is based on the maximum number of cases and items in analytics at any time, combined with the average size, and overheads accounting for conversations and row sizes.

So, the Discovery customer database also requires the following additional storage for the Analytics file groups:

Total capacity (MB) = 0.0166097i+0.00025ai

Where:

- i Total number of items (including attachments) in analytics at any time.
- a Average original item size.

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

The calculation also assumes the average converted content ratio, based on typical email and office attachments. If the source data contains more text-concentrated documents such as log or text files then the ratio is likely to be higher, or if the data contains more binary files such as images then the ratio is likely to be lower. This could be accommodated by changing the value of 0.00025 by either increasing or decreasing the value by 0.0001.

The following is an example medium scenario. Up to 25 cases that contain an average of 15,000 items per case (375,000 items in total) are expected to be in analytics across all Discovery cases at any time. The original items have an average size of 160 KB including attachments, with 20% of messages containing one or two attachments—therefore making up an extra 30% of items (approximately 487,500 items in total).

$$\begin{aligned}
 \text{Total capacity (MB)} &= 0.0166098i + 0.00025ai \\
 &= (0.0166098 * 487,500) + (0.00025 * 487,500 * 160) \\
 &= 8,097.28 + 19,500 \\
 &= 27,597.28 \text{ MB}
 \end{aligned}$$

So, the total capacity of all file group partitions needs to be 27.6 GB plus 20% extra headroom, which gives 33.12 GB.

Collecting analytics data for a single case is likely to produce sustained periods of very high I/O activity. This varies according to the available physical memory and other concurrent activities. On a database server that provides a throughput of up to 150,000 items per hour, the I/O activity might typically be 1,200 to 2,900 IOPS at the analytics case data file and 100 to 200 IOPS at the customer database log file. The process may take several hours to complete.

You can distribute the load from multiple analytics data collections by using multiple file group locations, which are defined when creating the Discovery customer database. Divide the estimated storage between at least as many independent partitions as there are expected to be simultaneous cases enabled for analytics. So, if multiple cases are concurrently performing analytics data collection, they should be using different partitions.

Note: Concurrent analytics data collection applies very high memory pressure, which is reflected in even higher I/O activity per case. See the memory requirements above.

Once the analytics data collection has completed, the normal I/O activity is much lower. The partition will be selected again for subsequent cases, but this should not have any impact on other cases that are still in analytics and located on the partition.

Anti-virus considerations

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

For more information, see Symantec technical article [TECH176828](#), “Recommended list of antivirus exclusions for SQL Server when used for Symantec Enterprise Vault, Compliance Accelerator and Discovery Accelerator”.

Virtualized infrastructure

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

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

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

Note the following general guidelines:

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

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

The disk partitions to be used for the database data files should be created as recommended by the hypervisor vendor for random access (most likely virtual hard disks).
- Virtual hard disks should be created as fixed size and not dynamic.
- 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 hyper-visor.
- The number of processor cores as recommended above should be exclusively dedicated to the virtual machine, and the processor priority and bandwidth set to provide the virtual machine with full utilization of the selected CPUs.

If you want to install the SQL Server instance on a virtualized machine, take care not to install multiple instances on the virtual machine.

Database tuning

The database server requires additional tuning to ensure that the best performance is achieved. In addition, as the Discovery Accelerator database grows, it requires regular maintenance, monitoring, and potentially tuning according to the usage patterns.

SQL permissions

The facility to create configuration and customer databases with Discovery Accelerator is dependent on the Vault Service account having the SQL Server role of database creator (dbcreator).

You may also want to assign the system administrator (sysadmin) role to the Vault Service account. If this is undesirable, however, you can assign a minimum set of permissions to the account, as described below.

Permissions required to create scheduled searches

Discovery Accelerator provides the facility to create schedules with which you can conduct searches repeatedly or at some future time. These schedules are SQL Server Agent jobs and, by default, Discovery Accelerator assumes that you want to make a user with the SQL system administrator (sysadmin) role the creator and owner of them. Therefore, you can either grant the sysadmin role to the Vault Service account or take the following additional steps to assign the minimum permissions needed to the Vault Service account:

- Add the Vault Service account to the msdb system database.
- Grant the Vault Service account Select permissions on the following msdb tables: sysjobs, sysjobschedules, sysjobsteps, and syschedules.
- Grant the Vault Service account Execute permission on the stored procedure sp_add_category.
- Assign the database role SQLAgentUserRole to the Vault Service account.

Permissions required to enable cases for analytics

To enable cases for analytics, the Vault Service account requires a number of permissions in various msdb tables and stored procedures. Without these permissions, errors occur when you try to enable a case for analytics.

Take the following steps to assign the minimum permissions needed to the Vault Service account:

- Add the Vault Service account to the msdb system database.
- Grant the Vault Service account Select permissions on the following msdb tables: sysjobhistory, sysjobs, sysjobschedules, sysjobservers, sysjobsteps, and syschedules.
- Grant the Vault Service account Execute permissions on the following stored procedures: sp_add_category, sp_add_job, sp_add_jobschedule, sp_add_jobserver, and sp_add_jobstep.
- Assign the database role SQLAgentUserRole to the Vault Service account.

Deploying databases and the impact of model database

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

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

- The database data file autogrowth value should be set according to the recommended values in the table below. This can be viewed using SQL Management Studio database properties, or the following SQL statement could be used to gather the sizes (in 8 KB pages):

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

Recommended auto growth values for databases

Database	Data file autogrowth value
Configuration	50 MB, unlimited growth.
Custodian	Approximately 1 – 2 days growth as per sizing guide. For example 50 MB, unlimited growth.
Customer	Approximately 1 – 2 days growth as per sizing guide. For example 100 MB, unlimited growth.

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

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

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

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

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

The tempdb database

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

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

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

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

- To prevent tempdb file growth causing unnecessary I/O, and to ensure that the tempdb files do not become severely fragmented, set the minimum file sizes to 200 MB and set them to grow by 10%.
- Move the tempdb data files to a dedicated striped disk array (RAID-0 or 10) and the log file to a dedicated high speed drive with high transfer rates (single disk or RAID-10). You can use the following SQL statement to move the file location:

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

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

- Create one data file per CPU, taking into account CPU affinity mask settings (to spread the load and reduce contention), making each file the same size. This can be achieved either using the SQL Server Management Studio or the following SQL statement:

```
USE master
ALTER DATABASE tempdb
ADD FILE (NAME = tempdev2, FILENAME = 'E:\SQLData\tempdb2.ndf',
        SIZE=200MB, FILEGROWTH=10%);
GO
```

- Do not replicate the tempdb storage device or configure SQL Server replication on the tempdb database. The tempdb database is a temporary database containing transient data that is recreated on each restart of SQL Server, so it will not need to be recovered.

Database maintenance

Implementing a Discovery Accelerator database maintenance plan is essential to preserve the database performance. As the database is used, crucial tables and their indexes fill and become fragmented. This reduces overall performance and leads to additional I/O when updating (due to page splitting). In addition, certain tables can significantly change profile each day, which results in out-of-date statistics, potentially causing inefficient query plans.

The solution to fragmented indexes is to rebuild very active indexes regularly. This can also help to reduce page splitting. However, some indexes can fragment very quickly due to excessive page splitting, such that rebuilds must be performed at an impractical frequency. In those cases, the index fill factors might need to be reduced to allow more free space per page, which prevents the excessive page splitting.

The best way to perform these optimizations is through monitoring, analysis, and tuning. The performance of indexes depends on the customer-specific data and their usage patterns. Rebuilding indexes can be time-consuming. There are a number of methods available depending on when the re-indexing can occur, and each provides different levels of improvements (see SQL Server books online for reorganizing and rebuilding indexes).

Reducing the fill factor is a balancing act between reducing page splits and associated I/O during updates while also ensuring the read performance is not significantly impacted by having to read many more pages and the associated I/O.

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

Table	Notes
tblSearchItems	Used during searching to store captured items before acceptance. Likely to have an equal or higher write-to-read ratio.
tblIntDiscoveredItems	Used to store all accepted items. Likely to have a higher read-to-write ratio.
tblIntAnalysedItems_<case#>	Used to store all original item content and other item metadata. During analytics case data collection, the related tables have a higher write-to-read ratio. Once this process is completed, the tables have a higher read-to-write ratio.

The statistics for the indexes on the following key tables are automatically updated on a regular basis to ensure optimum query plans, so they do not require any specific statistic maintenance.

- tblIntDiscoveredItems

■ tblSearchItems

■ tblAddressUser

■ tblDiscoveredItemToSearch

■ tblHistCasePermission

■ tblIntSearches

■ tblProductionToDiscoveredItem

■ tblItemSetItems

■ tblIntMarkHistory

■ tblCase

■ tblConversations_<case#>

■ tblContacts_<case#>
- tblPrincipal

■ tblAddress

■ tblVaults

■ tblHistCaseAddressUser

■ tblIntMessageCaptureSummary

■ tblIntSecurity

■ tblSearchSchedule

■ tblSearchResults

■ tblItemPolicy

■ tblIntAnalysedItems_<case#>

■ tblIntConversationItems_<case#>

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

Note: The database maintenance plan should not include a database or data file shrink. As for the above storage advice above, the database should have pre-allocated storage to avoid file growths and therefore any data file shrinking will be counterproductive.

Database upgrades

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

- Backup all Discovery Accelerator databases.
- Perform the SQL Server upgrade.
- Check the Discovery Accelerator database compatibility levels.
- Rebuild indexes.
- Update statistics with full scan.

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

- Backup all Discovery Accelerator databases.
- Perform the Discovery Accelerator database upgrades.
This will change the recovery model to “SIMPLE” and then return it to “FULL”.
- Rebuild indexes.
- Update statistics.
- Backup all Discovery Accelerator databases..

Advanced database tuning

There are certain circumstances in which the SQL Server may benefit from additional tuning, and there are also some more advanced approaches to improving database performance which may or may not be beneficial.

Multiple SQL Server instances

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

The SQL Server instance that hosts the Discovery Accelerator customer database should have the minimum memory option configured to ensure that the memory recommended in the “Memory considerations” section on page 12 is dedicated to it.

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

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

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

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

In the case of NUMA architectures, the processors, cores, and memory are arranged in nodes that should be carefully allocated to the SQL Server instances to avoid creating a memory or I/O bottleneck. It may be worth considering using Soft-NUMA to divide the resources with both non-NUMA and NUMA-based hardware. See the Microsoft Press books for more information.

Distributing I/O with multiple database files

Using multiple database files should not be necessary if the Discovery customer database files are placed on an adequately sized array.

Medium and large environments might consider creating multiple data files on separate striped disk arrays (RAID-10). This may improve query performance in large databases by enabling parallel queries in the different files.

Creating multiple data files causes SQL Server to distribute the database tables and indexes between the files, as determined by SQL Server. The use of multiple files lets SQL Server use multiple threads to access data distributed over those files, which may help improve performance by splitting sequential work into parallel activity.

SQL Server controls how the data is distributed between the files. Adding files to an existing large database may not necessarily produce any benefits. The number of files and their location also affects the performance. This depends on the I/O capabilities of the disk array on which the data is located, and placing multiple data

files on a single array may cause performance problems. Two main problems need to be avoided: I/O contention and file fragmentation.

I/O contention

Placing too many data files on a single disk array can cause a greater I/O load than the array can manage, which may slow performance. To ensure the throughput can be maintained, you may need to balance the data files between multiple disk arrays.

File fragmentation

File fragmentation can occur when multiple data files reside on a single partition, particularly if the data files do not have a suitable minimum size and so have to grow regularly. Each file growth ends up interleaved between the other data files and causes a significant performance impact. Therefore, each data file should be created with a large minimum size, with perhaps the combined total equivalent of one year's expected growth. Also ensure that the database is grown by a large fixed size rather than by percentage.

A general rule may be to create as many data files as there are CPUs, taking into account CPU affinity mask settings, and place them on separate disk arrays (RAID-10).

Moving Discovery Accelerator tables or indexes into file groups

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

Rolling over customer databases

You may need to roll over customer databases to remain within the available storage capacity. The database storage requirements are high. To reduce storage costs, you may want to roll over the database periodically to a new customer database (that is, create all new cases in a new database), and then archive the old database when it is no longer active. Alternatively, you can move the old customer database to slower storage and disable the associated Discovery Accelerator customer tasks. This would allow the customer database to be brought online quickly.

Note: You cannot consolidate customer databases. So, after you have created a case in a new customer database, you cannot merge it back into the old customer database.

Co-locating multiple customer databases

Multiple customer databases may be used within a Discovery Accelerator installation as an alternative method of logically separating the data, distributing the I/O load, keeping the database within acceptable sizes for performance, and enabling more concurrent activity between legal discovery users.

It is also possible that multiple Discovery and Compliance Accelerator installations may want to use the same database server.

Hosting more than one very active database on a single database server can become detrimental to performance, particularly if each Discovery customer database is expected to use analytics. You must size the database server appropriately to host the additional load, and the cost of scaling up a single server can become less cost-effective than using multiple database servers. (Scaling up will hit limitations, as well). In this case, consider multiple Discovery Accelerator installations with independent servers.

The customer databases may be divided by factors such as function (Litigation/HR), location, or even by individual legal discovery users. Typically, Litigation and HR customer databases are candidates for hosting on the same SQL Server due to the different use cases.

For example, litigation tends to be very active with large searches of journal archives, whereas HR-style customer databases tend to have low activity with very specific searches of individual mailbox archives.

Note: Additional customer databases may create an impact at the Discovery Accelerator server and the underlying Enterprise Vault infrastructure, which needs to be considered before implementation. For more information, see the chapter “Discovery Accelerator server” on page 43.

Monitoring the Discovery database server

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

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

Monitor the Discovery Accelerator database server during particular activities to ensure that the environment is performing correctly and allow appropriate database tuning.

Two activities to monitor for benchmark purposes are the following:

- When you accept a large search in an existing database with a large volume of results.
- When you enable a large case for analytics.

Remember that, in isolation, these activities do not represent peak load. In production use, the database is also under load from different activities such as search result processing, export processing, and various update activities. You can also use Windows Performance Monitor to obtain system and SQL Server statistics. Typically, you should monitor the following counters.

Object	Counters	Instances
PhysicalDisk (and potentially LogicalDisk as well)	Avg. Disk Read Queue Length Avg. Disk Write Queue Length Disk Transfers/sec Avg. Disk Bytes/Transfer Avg. Disk sec/Transfer Disk Bytes/sec Split IO/sec	SQL Data and log file drives
Memory	Page Faults/sec Pages/sec Available Bytes	
Processor	% Processor Time	_Total & All processors
System	Processor Queue Length Context Switches/sec	
SQLServer:Buffer Manager	Buffer cache hit ratio Page life expectancy Procedure cache pages Lazy writes/sec Checkpoint pages/sec	
SQLServer:Access Methods	Page Splits/sec Full Scans/sec	
SQLServer:Memory Manager	Total Server Memory (KB) Target Server Memory (KB)	

Object	Counters	Instances
SQLServer:Databases	Transactions/sec	_Total & Discovery Accelerator Database
SQLServer:SQL Statistics	Batch Requests/sec SQL Compilations/sec SQL Re-Compilations/sec	
SQLServer:Locks	Average Wait Time(ms) Lock Timeouts/sec Lock Waits/sec Number of Deadlocks/sec	_Total
SQLServer:Latches	Average Latch Wait Time(ms) Latch Waits/sec	

CPU and memory

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

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

In addition, if the context switches/sec are above 15,000 per CPU when you experience high CPU, it is possible that the server is spending too much time switching between threads of equal priority (but only if the CPU time is above 80%). This may occur for various reasons, as described in the Microsoft books. However, this is most likely to occur with other co-existing software such as multiple SQL Server instances. In this situation, see the section “Multiple SQL Server instances” on page 27.

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

Disk

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

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

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

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

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

SQL Server

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

Counter	Notes
Buffer Cache Hit Ratio	Should be above 90% to avoid too much I/O. A lower value may indicate too little server memory.
Total Server Memory, Target Server Memory	If the total server memory exceeds or is equal to the target server memory, there may be an issue with memory pressure. The server may require more memory.
Page Life Expectancy	Indicates how long pages remain in memory. Values that are regularly less than 300 seconds may indicate insufficient memory.
Lazy Writes/sec	Indicates how many times the lazy writer is moving changed pages to disk to free buffer space. This should be quite low. High values indicate high I/O, which more memory will help to reduce.

Counter	Notes
Page splits/sec	Ideally should be around or below 80 – 100 per second. Index fill factors can be examined to improve this situation.
Batch Requests/sec, Transactions/sec(_Total)	Can indicate the number of SQL requests, and therefore the overall load the SQL Server is handling.

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

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

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

Where you must replace `<database_name>` with the name of the Discovery Accelerator customer database.

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

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

It is essential to measure the index fragmentation for particular key tables, as described in “Database maintenance” on page 25.

In SQL Server 2005 onwards you can execute the following SQL statement. This outputs statistics on all tables and indexes where the external fragmentation exceeds 10% and the table consists of at least 1,000 pages.

```
SELECT OBJECT_NAME(i.object_id) AS TableName,
i.name AS TableIndexName, phystat.avg_fragmentation_in_percent
FROM sys.dm_db_index_physical_stats(DB_ID(), NULL, NULL, NULL,
'DETAILED') phystat
JOIN sys.indexes i ON i.object_id = phystat.object_id AND i.index_id =
phystat.index_id
WHERE phystat.avg_fragmentation_in_percent > 10 AND phystat.page_count >
1000
```

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

To do this in SQL Server 7 and 2000, execute the following SQL:

```
DBCC SHOWCONTIG (tblIntDiscoveredItems) WITH ALL_INDEXES
```

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

Useful queries

The following queries can be used on the Discovery customer database to identify search, acceptance, analytics, review marking, and export throughputs.

Identifying overall search throughput per hour

Execute the following query against each Discovery customer database. It outputs the number of hits retrieved, an estimated number of items searched, and the number of indexes searched per hour for the past 24 hours.

```
DECLARE @32bitExchMbx int,@64bitExchMbx int,@32bitExchJnl
int,@64bitExchJnl int,@32bitFSA int,@64bitFSA int,@32bitDominoMbx
int,@32bitDominoJnl int,@64bitDominoMbx int,@64bitDominoJnl int
SELECT
@32bitExchMbx=50000,@64bitExchMbx=50000,@32bitExchJnl=400000,@64bitExch
Jnl=5000000,@32bitFSA=60000,@64bitFSA=5000000,@32bitDominoMbx=50000,@64
bitDominoMbx=50000,@32bitDominoJnl=400000,@64bitDominoJnl=5000000
select left (convert (varchar, tblRate.enddate,20),14) as 'finished
date',count(distinct(searchid)) as 'searches running',
SUM(sitems) as 'estimated items searched',
sum(tblrate.numhits) as 'hits retrieved and processed',
```

Monitoring the Discovery database server

```

count(*) as 'index volumes processed',
avg(tblRate.duration) as 'avg search and processing duration(sec) per index
processed',
count(distinct indexserverid) as 'EV index services searched',
sum(distinct(totvaults)) as 'total index volumes included in running
searches'
from
(
select
a.searchid,y.totvaults,a.searchvaultid,a.startdate,a.enddate,a.numhits,
a.statusid,
"duration" = DATEDIFF(ss,a.startdate,a.EndDate),
"hitrate" = (Cast(a.NumHits as
float)/DATEDIFF(ms,a.startdate,a.EndDate))*1000*3600,
"sitems" = SUM(CASE
WHEN c.Type=9 AND ivs.VolumeType=0 THEN @32bitExchMbx
WHEN c.Type=9 AND ivs.VolumeType=1 THEN @64bitExchMbx
WHEN c.Type=17 AND ivs.VolumeType=1 THEN @64bitExchJnl
WHEN c.Type=17 AND ivs.VolumeType=0 THEN @32bitExchJnl
WHEN c.Type=129 AND ivs.VolumeType=1 THEN @64bitFSA
WHEN c.Type=129 AND ivs.VolumeType=0 THEN @32bitFSA
WHEN c.Type=1025 AND ivs.VolumeType=0 THEN @32bitDominoMbx
WHEN c.Type=1025 AND ivs.VolumeType=1 THEN @64bitDominoMbx
WHEN c.Type=513 AND ivs.VolumeType=1 THEN @64bitDominoJnl
WHEN c.Type=513 AND ivs.VolumeType=0 THEN @32bitDominoJnl
ELSE 0
END),
c.indexserverid
from
(select
searchid,searchvaultid,startdate,enddate,numhits,indexvolumesetid,vault
id,statusid from tblSearchVault z with (nolock)
union all
select
searchid,searchvaultid,startdate,enddate,numhits,indexvolumesetid,vault
id,statusid from tblSearchVaultsArchived x with (nolock)) a

```

```

        inner join tblVaults c with (nolock) on a.vaultid=c.vaultid
        inner join tblIndexVolumeSet ivs with (nolock) on
a.VaultId=ivs.VaultID and a.IndexVolumeSetID=ivs.IndexVolumeSetID

        left outer join (SELECT searchid,count(*) as 'totvaults' from
tblSearchVault with (nolock) group by searchid union all SELECT
searchid,count(*) as 'totvaults' from tblSearchVaultsArchived with
(nolock) group by searchid) y on a.searchid=y.searchid
where a.enddate is not null
and a.enddate > dateadd("hh", -24, getutcdate ())
group by
a.searchid,y.totvaults,a.searchvaultid,a.startdate,a.enddate,a.numhits,
a.statusid,c.indexserverid
) as tblRate
group by left (convert (varchar, tblRate.enddate,20),14)
order by "Finished Date" asc

```

Note the following:

- The query provides an estimated number of items searched based upon the average number of items in each type of archive. This may vary between environments and therefore should be tuned by changing the variable definitions at the start (if a value is unknown, the default should be used). Appropriate values can be obtained from the Enterprise Vault directory database using the following query:

```

SELECT --(CASE WHEN indexvolumetype=0 THEN '32-bit' WHEN
indexvolumetype=1 THEN '64-bit' END) AS 'VolumeType', -- EV 10 only
[9] AS 'Exchange Mailbox', [17] AS 'Exchange Journal', [33] AS 'Public
Folders', [65] AS 'SharePoint', [129] AS 'FileSystem',[257] AS 'WSS',
[513] AS 'Domino Journal', [1025] AS 'Domino Mailbox'
FROM (select --indexvolumetype, -- EV 10 only
Type,IndexedItems from IndexVolume iv inner join root a on
iv.RootIdentity=a.RootIdentity) AS SourceTable
PIVOT (AVG(IndexedItems) FOR Type IN
([9],[17],[33],[65],[129],[257],[513],[1025])) AS PivotTable

```

Note: The commented rows [--] above should be uncommented if the query is to be executed on an Enterprise Vault 10.0 directory database.

- This query outputs the throughput per hour across all searches rather than an average throughput per search. This is because multiple searches can be executed concurrently, and the overall throughput is shared between the

searches. Therefore, the individual search throughputs could be very misleading.

- The query only outputs rows for time periods in which index searches completed, so the results may exclude time periods.
- The number of hits retrieved per hour may vary considerably, as this depends on whether the searches conducted during this period actually found any hits. The number of indexes searched per hour varies depending on how many searches have been conducted in this time period, how many results were retrieved and how many archives (and their indexes) are included in each search.

The following is sample output from searches across thousands of archives.

Finished date	Searches running	Estimated items searched	Hits retrieved and processed	Index volumes processed	Avg search and processing duration(sec) per index processed	EV index services searched	Total index volumes included in running searches
2011-10-07 10:	1	2,931,104	146,927	6,056	7	8	15,992
2011-10-07 11:	2	12,547,700	1,015,456	25,925	8	8	31,992
2011-10-07 12:	2	10,164	5,370	21	7	7	16,010

Identifying overall search acceptance throughput per hour

Execute the following query against each Discovery customer database. It outputs the number of items accepted per hour for the past 24 hours.

```
select left (convert (varchar, capturedate,20),14) as 'Accept Date',
        count(*) as 'Items Accepted in hr'
from tblIntDiscoveredItems with (nolock)
where capturedate > dateadd("hh", -24, getutcdate ())
group by left (convert (varchar, capturedate,20),14)
order by "Accept Date"
```

Note the following:

- This query only outputs rows for time periods in which items were accepted, so the results may exclude time periods.
- The number of items accepted per hour may vary depending on how many searches were accepted in a given hour and how many items were included in each search.

The following is sample output:

Accept Date	Items Accepted/hr
12/03/2009 15:	199991
12/03/2009 16:	137817
12/03/2009 17:	1462274
13/03/2009 09:	200029

Identifying export throughput per hour

Execute the following query against each Discovery customer database. It outputs the number of items exported per hour for the past 24 hours.

```
select left (convert (varchar, tblRate.completiondate,20),14) as 'Finished
Date',
sum(tblrate.numitemsproduced) as 'items exported by exports completed
within hour',
avg(tblrate.numitemsproduced) as 'average items exported per export
completed within hour',
avg(tblRate.duration) as 'avg duration (s)/export',
avg(tblRate.exprate) as 'avg items exported per export/hr',
avg(tblRate.overlapping) as 'avg overlapping exports',
avg(tblRate.exprate)*(avg(tblRate.overlapping)+1) as 'estimated overall
export throughput/hr'
from
(
select a.productionid,a.numitemsproduced,b.name,
"duration"=datediff(ss,a.createdate,a.completiondate),
"exprate"=(Cast(a.numitemsproduced as
float)/DATEDIFF(ms,a.createdate,a.completiondate))*1000*60*60,
a.CreateDate,
"overlapping" = (select count(*) from (SELECT b.productionid from
tblProduction b with (nolock) where (b.completiondate>a.createdate AND
b.completiondate<a.completiondate) union select d.productionid from
tblProduction d with (nolock) WHERE (d.createdate>a.createdate AND
d.createdate<a.completiondate) ) as tblSubQ),
a.completiondate
```

```
from tblProduction a with (nolock)
inner join tblstatus b on a.statusid=b.statusid
where a.completiondate is not null
and a.completiondate > dateadd("hh", -24, getutcdate ())
) as tblRate
group by left (convert (varchar, tblRate.completiondate,20),14)
order by "Finished Date" asc
```

Note the following:

- This query only outputs rows for time periods in which items were exported, so the results may exclude time periods.
- The number of items exported per hour may vary depending on how many exports have been conducted in a given hour and how many items were included in each export.

The following is sample output:

Finished Date	Items exported by exports completed within hour	Average items exported per export completed within hour	Avg duration (s)/export	Avg items exported per export/hr	Avg overlapping exports	Estimated overall export throughput/hr
13/03/2009 12:	20000	20000	190	378901	0	378901
13/03/2009 20:	20000	5000	393	46704	2	140114

Identifying review marking throughput per hour

Execute the following query against each Discovery customer database. It outputs the number of items marked per hour for the past 24 hours.

```
select left (convert (varchar, markdate,20),14) as 'Mark Date',
count(*) as 'Total Items Marked in hr'
from tblIntMarkHistory with (nolock)
where markdate > dateadd("hh", -24, getutcdate ())
group by left (convert (varchar, markdate,20),14)
```

Note the following:

- This query only outputs rows for time periods in which items were marked, so the results may exclude time periods.
- The number of items marked per hour may vary depending on how many reviewers were working in a given hour, their working practices, and therefore how many items were marked.

The following is sample output:

Mark Date	Total Items Marked/hr
27/03/2009 08:	26238
27/03/2009 09:	26357

Identifying overall analytics ingestion throughput per hour

Execute the following query against each Discovery customer database. It outputs the number of items retrieved from Enterprise Vault and inserted into the analytics tables per hour for the past 24 hours.

```
DECLARE @sCmd nvarchar(1000)
DECLARE @curTable int, @LastTable int
DECLARE @tbls TABLE (rownum int IDENTITY(1,1) NOT NULL, TableName
nvarchar(255) NOT NULL, OwnerName nvarchar(255) NOT NULL )
CREATE TABLE #Results (UpdatedDate varchar(255), MsgAttHr int, MsgHr int,
MsgAttSizeHr int)
INSERT @tbls (TableName, OwnerName) SELECT object_name(o.id), u.[name]
FROM sysobjects o INNER JOIN sysusers u ON u.uid = o.uid WHERE o.type='U'
and o.[name] like 'tblIntAnalysedItems%'
SET @LastTable = @@ROWCOUNT
SET @curTable = 0
WHILE (@curTable < @LastTable )
BEGIN SET @curTable = @curTable + 1
SELECT @sCmd = N'select "UpdatedDate" = left
(convert (varchar, lastupdated,20),14),"MsgAttHr" = count
(*),"MsgHr"=sum(case when parentrowid is null then 1 else 0
end),"MsgAttSizeHr" = sum (size) from ' + TableName + N' with (nolock) where
lastupdated is not null and lastupdated > dateadd("hh", -24, getutcdate ())
group by left (convert (varchar, lastupdated,20),14) order by "UpdatedDate"
asc'
FROM @tbls
WHERE rownum = @curTable
INSERT #Results EXEC sp_executesql @sCmd
END
SELECT [Ingest Date]=UpdatedDate,
[Messages and attachments/hr]=SUM(MsgAttHr),
[Top level messages/hr]=Sum(MsgHr),
[Messages and attachments size(kb)/hr]=Sum(MsgAttSizeHr)
FROM #Results a
```

```
Group By a.UpdatedDate
ORDER BY a.UpdatedDate ASC
DROP TABLE #Results
GO
```

Note the following:

- This query only outputs rows for time periods in which items were ingested, so the results may exclude time periods.
- The number of items ingested per hour may vary depending on how many cases were concurrently enabled for analytics, and the size of each case.

The following is sample output:

Ingest Date	Messages and attachments/hr	Top level messages/hr	Messages and attachments size(kb)/hr
14/04/2009 13:	56788	43715	10032449
14/04/2009 14:	166727	127982	29599214
14/04/2009 15:	139046	106917	25096562
14/04/2009 16:	129148	99256	22630495

Discovery Accelerator server

The Discovery Accelerator server directs the flow of data between Enterprise Vault and the Discovery Accelerator database, and manages the interaction with end-users. End-user actions usually result in Discovery Accelerator database activity, but may also involve Enterprise Vault.

The Discovery Accelerator server can potentially be very resource-intensive, and so it may not best co-exist with other applications or services. In a small environment, the server can potentially co-exist with the Enterprise Vault services, but you need to pay careful attention to the hardware specification.

In most situations, you should not host other applications or services with Discovery Accelerator. The Discovery Accelerator server may need to be scaled out, depending on the expected use.

Hardware requirements

The Discovery Accelerator service concurrently coordinates significant activity and therefore benefits from multi-processor servers. The flow of data through the Discovery Accelerator service can cause high memory utilization, and a multi-user environment can cause the customers' service to consume resources. In addition, the Discovery Accelerator pre-fetch cache and export features can result in very high I/O loads.

We recommend that the Discovery Accelerator database server and application server are connected through gigabit network technology. It would also be highly beneficial if the Enterprise Vault infrastructure can be made available to Discovery Accelerator through gigabit technology.

Server considerations

The Discovery Accelerator service creates many threads of execution to manage different concurrent activities. A multiprocessor server is essential to ensure that all these concurrent activities can function with reasonable performance. A small environment may require Discovery Accelerator to co-exist with Enterprise Vault. If co-existing with Enterprise Vault 9.0, at least four physical cores are recommended. With Enterprise Vault 10.0, at least eight physical cores are recommended.

All other installations should have a dedicated Discovery Accelerator server with at least four processor cores available to the operating system. Either four physical CPUs or a combination of multi-core and multi-CPU can be used, but server sizing must not be based upon hyper-threading. If analytics or more than one customer database is to be used then at least eight processor cores should be installed.

The Discovery Accelerator service can require a high memory capacity, particularly during searching and analytics data collection.

A 64-bit Discovery Accelerator server should have 8 GB of RAM available to the operating system to handle the high memory requirements of Discovery Accelerator. Discovery Accelerator still requires the use of 32-bit processes that individually have limited memory. However, the available memory is utilized between the multiple processes. If multiple customer databases are to be used, the memory should be increased by at least 4 GB per active customer database.

The Discovery Accelerator server may need to be scaled out, as discussed in the following sections, to balance the load between customer tasks, reviewing, and analytics data collection. Additional servers for customer tasks and Analytics purposes should be sized as above.

If the online preview and reviewing facilities are expected to host more than 300 concurrent reviewers, the Discovery Accelerator customers' service should be scaled out to additional servers in a network load-balanced arrangement. Each additional server can consist of a 2-core server with 6 GB of RAM, which should typically support up to 300 lightweight reviewers. However, this depends on the working practices of those users, and more active users may require 8 GB of RAM and a 4-core server.

When scaling out Discovery Accelerator, each customer database requires a customer tasks service, each of which can only reside on one server.

Storage considerations

The Discovery Accelerator service that runs the customer tasks requires storage to be arranged for two different purposes: the pre-fetch cache, and the export location. By default, the pre-fetch cache is limited to 1 GB disk capacity, but you may need to increase this to make the best use of it. For more information, see “Pre-fetch cache tuning” on page 50.

The export location storage requirements vary depending on the total number of anticipated exports at any one time and the total size of all the original items for each export. Additional space is required for overheads during processing of Exchange message production to PST. The storage needs to be nearly double the total export size during processing. However, once complete, the additional space is released.

A single export to native format generally produces 200 – 300 IOPS. However, exporting to Exchange PST incurs overheads which can produce between 900 and 1,600 IOPS. In addition, concurrent exports can increase this load considerably. Therefore, it may be worth providing multiple export locations to distribute the I/O load. For more information, see “Export and production tuning” on page 54.

The type of storage and interfaces that you use must ensure that the storage for either of these purposes does not become a bottleneck. LAN-based storage should not be used for the pre-fetch cache and may not be suitable for the export location. The best devices are local storage, direct attached storage, or partitions on appropriately sized storage area network (SAN).

Both the pre-fetch cache and export location require good random access or a high number of IOPS, so a striped array of many disks should be used (using hardware-based RAID rather than software-based RAID). Due to the size and volume of data involved with these activities, the RAID also requires good transfer rates and therefore high speed disks should be used. These two sets of files need to be located on different arrays to ensure that the export throughput and pre-fetch cache benefits 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. However, writing incurs a cost of four to six physical writes per operation. This may be acceptable for the pre-fetch cache location, which should have a higher proportion of reads, but a poorly sized RAID-5/6 implementation can significantly reduce the performance of write-intensive exports. Correctly sizing a RAID-5/6 implementation to maintain write performance may become more costly than RAID-10, and therefore a RAID-10 scheme must be considered for the export location.

Redundancy may not be an issue for the pre-fetch cache, which contains a copy of the original data. Therefore, no data loss occurs if the cache is lost. However, only new items are added; the missing items are not repopulated unless Discovery Accelerator is instructed to rebuild the cache.

In the case of local or direct attached storage, multiple controllers supporting multiple channels should be used to distribute the load between the pre-fetch cache and export locations and provide sufficient throughput. The controllers should also provide a battery-backed read and write cache to aid performance.

When you select storage, consider the advice in "Storage" on page 10, and create and align partitions as advised.

A Discovery Accelerator server should typically have the following partitions:

- System drive (RAID-1)
- Pre-fetch cache (RAID-5 or RAID-10 striped array with many disks)
- One or more export locations (RAID-10 striped array with many disks)

Note: Exclude the export locations from anti-virus scanning. Otherwise, file locking issues may prevent the export from completing.

Virtualized infrastructure

When you install Discovery Accelerator in a virtualized infrastructure, there are important aspects to consider. Follow the recommendations of your hypervisor vendor when you size and configure the environment. In addition, you must ensure that the resource requirements described above are dedicated to the virtual machine to minimize the impact on performance from intermediate layers or co-existing guests.

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

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

The disk partitions to be used for the pre-fetch cache and export location should be created as recommended by the hypervisor vendor for random access (most likely virtual hard disks).

- Virtual hard disks should be created as fixed size and not dynamic.
- The memory requirements recommended above should be dedicated and prioritized to the virtual machine to prevent dynamic allocation or sharing.

- Avoid the use of hyper-threading by the hyper-visor.
- The number of processor cores as recommended above should be exclusively dedicated to the virtual machine, and the processor priority and bandwidth set to provide the virtual machine with full utilization of the selected CPUs.

Custodian Manager tuning

You can install a single Custodian Manager database in a Discovery Accelerator installation. However, all Discovery Accelerator customer databases in the installation share the Custodian Manager database.

The Custodian Manager is managed through an IIS Web site, which is automatically installed on the server on which the Custodian Manager is created. This Web site has a low user load and should not have any impact on other activities on the server.

The Custodian Manager customer tasks service typically has little impact on the other Discovery Accelerator services, and requires little specific tuning. However, a large customer that expects any co-existing customer databases to be under heavy load may need to scale out the Custodian Manager to a dedicated server. In this situation, it is likely that the customers' service and analytics' service need to be scaled out as well, and therefore the Custodian Manager could be co-located on one of those servers.

Discovery Accelerator customers service tuning

The Discovery Accelerator customers service provides the client access service. Client/server communications are encrypted and digitally signed, by default using port 8086. The user token from the client workstation can only be used by the server for identification and not impersonation.

The customers service benefits from tuning to the specific environment requirements. If the online preview and reviewing facilities are to be used in large environments, the customers service should be hosted on a separate server.

Each customers service should support up to 300 concurrent users. If you need to support more than 300 users, add additional servers with appropriate Network Load Balancing implemented to direct the traffic between the servers.

Network sockets

You must adjust the Windows TCP/IP socket parameters to provide the .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	Default	Recommended (decimal)
TCPTimedWaitDelay	DWORD	120	60

3. Extend the dynamic port range using the following netsh commands to change the start port to 1025 (default 49152) and number of ports to 64510 (default 16384):
 - `Netsh int ipv4 dynamicport tcp start=1025 num=64510`
 - `Netsh int ipv4 dynamicport udp start=1025 num=64510`

Note: Setting the MaxUserPort registry key to 65535 will revert to pre-Windows 2008 functionality and set the start port to 1025 and number of ports to 64,510. However, it is recommended to use the netsh commands.

Discovery Accelerator analytics service tuning

The Discovery Accelerator analytics service benefits from tuning to the customer’s specific environment. Host this service on one or more additional servers if one or both of the following applies:

- You are likely to enable cases for analytics on a regular basis.
- You anticipate that you may want to enable several cases for analytics concurrently (such as medium or large customers).

Scaling out the Discovery analytics service improves performance and prevents contention with core Discovery activity. However, concurrent analytics data collection should be avoided where possible due to the impact at the database server. For more information, see the “Database platform” chapter on page 11.

Retrieval thread tuning

Even with a well-specified database server, more than six active Enterprise Vault Content Management (ECM) retrieval threads are ineffective due to the performance of inserting the data into the database. In addition, by default, an individual source Enterprise Vault server is limited to using up to two available threads. So, if the case search results to be ingested are from fewer than three source Enterprise Vault servers, analytics may not take advantage of the potential throughput. The per-server thread limit prevents an individual Enterprise Vault server from becoming overloaded with retrievals.

If the additional load upon individual Enterprise Vault servers is of no concern, it is beneficial to set both the maximum number of ECM threads and the thread limit per Enterprise Vault server to six. This ensures that however many source Enterprise Vault servers are required to service the ingestion, the optimum throughput is achieved.

However, a poorly specified database server impacts the performance such that six threads may not meet expectations.

Change the following Analytics Data Collections settings in the System Configuration:

Name	Default	Recommended
Maximum number of threads for fetching data from Enterprise Vault servers (hidden)	10	6
Maximum number of threads per vault server to retrieve data from Enterprise Vault (hidden)	2	6

Discovery Accelerator customer tasks tuning

The Discovery Accelerator service benefits from tuning to the customer’s specific environment. The underlying Enterprise Vault infrastructure and expected patterns of activity can impact the overall Discovery Accelerator performance.

Each customer database and the Custodian Manager database has its own customer tasks. You may need to scale these out if the individual customer databases are each under load from a high volume of searches and exports.

Pre-fetch cache tuning

The pre-fetch cache runs according to a nightly schedule to download original items from the Enterprise Vault storage service for all recently accepted search results to the server running the customer tasks. This can impact the performance of the Enterprise Vault services during the scheduled period, so it needs to be carefully considered before implementation.

The pre-fetch cache is used by the online preview and reviewing facilities and the export and production features. Retrieving original items from the cache can perform significantly better than from the Enterprise Vault storage service, and it can prevent the storage service from being heavily utilized during peak times of activity.

The pre-fetch cache mechanism can download two parts of the original item: the HTML version displayed for reviewing, and the native format used for exporting. By default, the service only downloads the HTML component for reviewing purposes.

Normally, it is not worth enabling the native format download for export purposes. This is because the cache only retains items for five days, and exports are unlikely to occur within this timeframe. Furthermore, the additional storage overhead results in less capacity being available for the HTML items for review.

During the pre-fetch cache update window, the pre-fetch mechanism downloads 200 items per 30 seconds by default. This prevents overloading on smaller systems, but on larger systems the batch size may need to be increased. This depends on how many items can be downloaded within 30 seconds. If the download takes just over 30 seconds, the software waits until the next 30-second interval before starting to download again. Therefore, the batch size may need to be adjusted to between 200 and 1000 to gain the best throughput.

Change the following Item Pre-fetch Cache settings in the System Configuration:

Setting	Default	Recommended
Cache enabled	Off	On
Cache location	—	Dedicated striped array.
Start prefetching time of day	20:00	Set time to avoid other scheduled activity.
End prefetching time of day	05:00	Set end time to prevent overrunning.
Cache maximum size (Mbytes)	1000	Sufficient to include 5 days of search results. This can be determined by monitoring the event log which will indicate when the cache is full.

Change the following Item Pre-fetch Cache (Advanced) settings in the System Configuration:

Setting	Default	Recommended
Prefetch Native format	Off	Off
Item batch size (hidden)	200	Value to match throughput during 30 seconds.

Search tuning

Discovery Accelerator searches by default up to ten index volumes concurrently per Enterprise Vault indexing service. Discovery Accelerator has a customer-wide soft-limit of 100 search threads, at which point the threads are then allocated equally between all Enterprise Vault index servers searched up to 100 index servers. Beyond this, each Enterprise Vault server is then allocated one thread.

Enterprise Vault 10.0 introduced a 64-bit indexing engine, and it continues to support any existing 32-bit indexes. In addition, Enterprise Vault 10.0 also introduced index groups to provide a distributed and scalable indexing architecture.

Discovery Accelerator 10.0 supports searching 32-bit indexes available in Enterprise Vault 9.0 and 10.0, and 64-bit indexes available in Enterprise Vault 10.0.

Depending on the Enterprise Vault server specifications and type of indexes, the default search thread settings may not be optimal. An improper value can potentially overload the indexing servers and storage devices, and the associated contention can significantly degrade the performance.

The 32-bit and 64-bit indexing engines have different performance characteristics and may require different tuning. The Discovery Accelerator search threads are a customer-wide setting regardless of index type or location. Therefore, if the Enterprise Vault servers contain a mixture of 32-bit and 64-bit index volumes, it may be beneficial to consider upgrading the 32-bit index volumes that will be searched to take full advantage of the 64-bit indexing engine.

See "Enterprise Vault indexing service considerations" on page 65 for more information.

You may need to adjust the number of search threads to balance the load evenly or take better advantage of well specified servers and dedicated index groups.

Increasing the number of search threads per Enterprise Vault server might help increase the index volume iteration rate, provided that at each Enterprise Vault server there is sufficient memory and the indexing storage does not become a bottleneck.

The optimum value for the number of threads depends on the specification of all Enterprise Vault index servers. An environment containing a mixture of different specification Enterprise Vault servers may need to be tuned to the average specification.

To determine the best value, make several benchmark searches representing typical anticipated searches. This may consist of a few large journal archives, or a large number of mailbox archives, or a combination of the two. Repeat the same search with different numbers of threads, starting with a low value such as two threads and increasing in steps of five. Use the SQL query in “Identifying overall search throughput per hour” on page 35 to measure the performance.

When an index volume is searched Discovery Accelerator will conduct any other searches queued for that index volume to make efficient use of the index whilst in memory. The number of consecutive searches needs to be balanced to ensure reasonable sharing between searches of other index volumes and to prevent resource contention. Environments containing 32-bit index volumes should limit the number of consecutive searches.

Change the following Search setting in the System Configuration, and then restart the customer tasks:

Setting	Default	Recommended
Number of Vault search Threads	10	32-bit only (Enterprise Vault 9.0): Start at 2 and tune to environment. 64-bit and mixed 32-bit/64-bit: 10 64-bit in dedicated index groups: Start at 10 and tune to environment
Maximum Number of consecutive searches on same index	0	32-bit and mixed 32-bit/64-bit: 100 64-bit only: 0

If you increase the number of search threads above 10, you may need to increase the Enterprise Vault 32-bit index server process limits on every Enterprise Vault server. However, you must take the server specification into consideration, as described in "Enterprise Vault indexing service considerations" on page 65.

If you run Enterprise Vault 9.0, increasing the number of search threads above 10 is only practical if all Enterprise Vault servers are running on 64-bit operating systems, with a minimum of 4 CPUs and 8 GB of RAM.

Add the following registry keys to each Enterprise Vault server hosting 32-bit index volumes, and then restart the indexing service:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\KVS\Enterprise Vault\Indexing
```

Setting	Default	Recommended
ThresholdIndexServers	20	10 + Number of vault search threads (but no fewer than 20).
MaxIndexServers	30	20 + Number of vault search threads (but no fewer than 30).

To optimize search result processing, ensure that the following Search setting in the System Configuration is enabled:

Setting	Default	Recommended
Use sequence numbers for searches	On	On

If you can use date ranges within searches, you can further improve the efficiency of the searches by enabling more selective searches that exclude indexes that do not contain documents within the given date range. To take advantage of this feature, enable the setting below. This setting does not have any impact on searches that you conduct without specifying a date range. Such searches continue to search all related index volumes.

This feature relies upon directory synchronization with Enterprise Vault to update the Discovery Accelerator database with date range details for each index. This occurs by default every six hours. However, if directory synchronization is disabled, or the update interval is increased beyond one day, there is a risk that a search may miss results if an index contains newer entries than the date range records in the Discovery database.

To enable date range optimization, change the following Search settings in the System Configuration. However, ensure that the directory synchronization is enabled and the update interval is at least once per day.

Setting	Default	Recommended
Optimise searches based on oldest and youngest items	Off	On

The Discovery Accelerator search page provides a feature-rich search that lets you find documents that are based on a large number of criteria. However, the search allows an open search that will retrieve every item in every index searched. Therefore, end-users should be advised to ensure searches are as specific as possible, particularly using date ranges if possible. Mandating fields can be an effective method of ensuring that no open searches are triggered (see the Search settings called “Require <field> to be specified” in the System Configuration).

Export and production tuning

Each export or production is allocated multiple threads. Each of the threads downloads original items either from the pre-fetch cache, if available, or directly from the Enterprise Vault storage services. The use of multiple threads prevents Discovery Accelerator from waiting for any slower Enterprise Vault storage services, and provides optimal throughput for a single export across multiple Enterprise Vault servers.

Discovery Accelerator 10.0 introduces a customer-wide default limit of 100 export threads. The limit can help prevent the Discovery Accelerator server from being overloaded by multiple concurrent exports. By default, each export starts 25 threads, which permits up to four exports to proceed concurrently.

Both limits can be adjusted to improve throughput or concurrency and/or reduce the effects of export on other Discovery Accelerator customer functions or the Enterprise Vault storage services. The thread limit must be divisible by the per-export thread setting.

The defaults are likely to be too high for many environments, and should be lowered to 10 threads with a limit of 20 threads (limiting to two concurrent exports) and then tuned according to the factors below.

To determine the best thread values for your devices, make several benchmark exports representing typical anticipated exports in order to do the following:

- Ensure that the export destination disks can sustain the load. Adjust the thread limit and potentially adjust the threads per export to allow the desired number of concurrent exports within the thread limit.
- Ensure that the system does not spend most of its time idle waiting for the Enterprise Vault storage services. If many threads are waiting then the threads per export may need to be increased. The thread limit may also need to be adjusted to accommodate the threads per export.
- Ensure better coexistence with searching and end-user activity by reducing CPU utilization. Adjust the thread limit and potentially adjust the threads per export to allow the desired number of concurrent exports within the thread limit.
- Increase the number of exports that can be run concurrently. It is important to consider that all exports share the overall system performance, so, by enabling more concurrent exports, each individual export will take longer to complete. To determine the number of threads to allocate to each export, divide the thread limit by the required number of concurrent exports.

Take care to ensure that the system is not tuned for infrequent peak loads, which could result in reducing the throughput during normal usage.

Change the following setting in the System Configuration:

Setting	Default	Recommended
Number of Production Threads Per Production Run	25	10 or based upon hardware and activity profile.
Total Number of Production Threads Per Customer	100	50 or based upon hardware and activity profile and divisible by the number of production threads above.

Architecture considerations

Medium and large customers may have distributed their Enterprise Vault infrastructure over multiple sites with varying levels of interconnectivity. In addition, Discovery Accelerator may need to be used by different parts of an organization located across different sites.

The legal discovery functionality is available through the Discovery Accelerator customers' service, which only generates low levels of traffic between the server and Discovery client, and therefore should allow the server to be utilized throughout the organization.

However, any exported data will tend to be very large, and in most situations it will be located on the Discovery Accelerator server. Transporting these large volumes of data across inter-site network links may become an issue. Multiple Discovery Accelerator installations may therefore be needed with additional Discovery servers behind slow or expensive network links.

The majority of work occurs between the Discovery Accelerator server and database server (which should be co-located within a site) and the Enterprise Vault infrastructure, which may be distributed.

The traffic between Discovery Accelerator and Enterprise Vault varies depending on the size of the Enterprise Vault implementation and the activity patterns of the legal discovery team. Background activities such as Enterprise Vault and Active Directory synchronization generate short and infrequent periods of high traffic, depending on the number of mailboxes.

The main source of traffic is during export, when original archived data is downloaded from storage services, and during searching, when results are returned from indexing services. The search result data is transported as XML, but the export data is transported as large volumes of binary data.

These activities may cause issues between sites with slow or expensive network links, and therefore may warrant local Discovery Accelerator servers to split the load. This would mean that a search of the entire organization may require the use of several different installations, but it is likely that most searches would be local to the site in which it resides.

Note: Discovery Accelerator can search for and retrieve data (for review, export, and analytics) across independent Enterprise Vault installations.

Using multiple customer databases

You can use multiple customer databases to logically separate the data. This is an alternative method of distributing the I/O load, keeping the database within acceptable sizes for performance, and enabling more concurrent activity between legal discovery users. This approach also has the advantage of sharing the Custodian Manager service between all of the customer databases. There is no need to duplicate custodian information between each customer database.

Small or medium environments may consider this approach to be a more manageable method of scaling the Discovery Accelerator database. Larger environments may want to consider multiple Discovery Accelerator installations due to the impact at the database server.

The customer databases can be divided by factors such as function (Litigation/HR), location, or even by individual legal discovery users. When considering hosting

multiple databases on a single SQL Server, pay careful attention to the expected level of activity. Hosting two very active databases on a single database server can become detrimental to performance, especially when analytics is used. In this case, you should consider multiple Discovery Accelerator installations.

Typically, Litigation and HR customer databases are candidates for hosting on the same SQL Server due to the different use cases. For example, litigation tends to be very active with large searches of journal archives, whereas HR-style customer databases tend to have low activity with very specific searches of individual mailbox archives.

However, adding customer databases increases the load at the Discovery Accelerator server, particularly if analytics is to be used. Each customer database has its own dedicated customer tasks service and analytics service. This can result in contention at the Discovery Accelerator server if the customer databases are heavily loaded with searches, exports and analytics data collection. In this situation, the customer task and analytics service for each customer database may need to be scaled out to separate servers.

You also need to pay careful attention to which Enterprise Vault servers are used by the separate customer databases. In the situation where multiple customer databases can access the same set of Enterprise Vault servers, the infrastructure must be capable of supporting the potentially high-load overlapping activity.

Each customer database can independently search and retrieve large volumes of data for review, export, and analytics. This may cause excessive resource utilization at the Enterprise Vault servers. It is essential that the customer databases balance the load by reducing the search and export threads, as described previously. The number of ECM retrieval threads may need to be reduced in each analytics service.

Using multiple Discovery installations

You may require multiple Discovery Accelerator installations in instances where multiple Enterprise Vault sites are installed in physically disparate locations, perhaps with slow or no network links or across boundaries of legal jurisdiction. Also, as the demands of the Discovery Accelerator environment grow, the volume of data and levels of activity between Discovery users may affect performance.

You may be able to achieve the following by installing multiple independent Discovery Accelerator installations on independent hardware:

- Meeting topological or legal boundaries.
- Distributing the load.
- Keeping each environment within acceptable sizes for performance.
- Enabling more concurrent activity between legal discovery users.

However, this has the disadvantage that each installation requires its own Custodian Manager. This decentralizes custodian management and increases the hardware requirements.

This approach may be required by medium or larger environments to scale out Discovery Accelerator, or to meet geological and legal boundary demands.

You can divide the separate installations by factors such as department function (Litigation/HR), geographical location, or even by individual legal discovery users. Each environment requires its own set of dedicated servers, including a database server and Discovery Accelerator server.

Pay careful attention to which Enterprise Vault servers can be used by the separate Discovery Accelerator installations. In the situation where multiple Discovery Accelerator installations are able to access the same set of Enterprise Vault servers, the infrastructure must be capable of supporting the potentially overlapping high activity.

Each Discovery Accelerator environment can independently search and retrieve large volumes of data for review, export, and analytics. This may cause excessive resource utilization at the Enterprise Vault servers. It is essential that the Discovery Accelerator services balance the load by reducing the search and export threads as described previously. The number of ECM retrieval threads may need to be reduced in each analytics service.

Monitoring Discovery Accelerator

Monitor the Discovery Accelerator server during the following activities to ensure that the environment is performing correctly:

- During a large export to native format (for example 100,000 results).
- During a search that includes many different journal style vaults (a large volume).
- During analytics case data collection.

Use Windows Performance Monitor to obtain system statistics. You should typically monitor the following counters:

Object	Counters	Instances
PhysicalDisk (and potentially LogicalDisk as well)	Avg. Disk Queue Length	Export location drives
	Disk Transfers/sec	
	Avg. Disk Bytes/Transfer	
	Avg. Disk sec/Transfer	
	Disk Bytes/sec	
	Split IO/sec	
Server	Files Open	
Memory	Page Faults/sec	
	Pages/sec	
	Available Bytes	
	Free System Page Table Entries	
Processor	% Processor Time	_Total
Network Interface	Output Queue Length	All instances
	Packets/sec	
System	Processor Queue Length	
	Context Switches/sec	
	File Data Operations/sec	
	Processes	

CPU and memory

The % Processor Time for the _Total counter indicates overall system activity. If the CPU is consistently above 80%, the server may start to indicate performance issues. If the % Processor Time is generally above 80%, and the Processor Queue length is generally above twice the number of CPUs, then the CPUs are likely to be a bottleneck.

In addition, if the context switches/sec are above 15,000 per CPU when you experience high CPU, it is possible that the server is spending too much time switching between threads of equal priority (but only if the CPU time is above 80%). This may occur for various reasons, but it is most likely to occur when there are too

many threads running concurrently. For example, the server may be overloaded with many concurrent exports with too many threads (see “Export and production tuning” on page 54), combined with other activities such as searching and on-line reviewing (consider scaling out the customers service).

It would be worth monitoring the memory counter Pages/sec, which records the number of hard faults to disk. If there are significant or sustained hard faults, trace the problem to the source.

Disk

The disk counters are only relevant if the pre-fetch cache and export file locations are locally attached. If the export file locations are on a remote server, monitor that server for disk activity.

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

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

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

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

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

The Server counter Files Open can also be useful to indicate how many files are in use at any time. This can help to indicate how many concurrent exports are occurring.

Discovery Collector

Discovery Collector identifies and gathers a wide range of electronically stored information (ESI) throughout an enterprise infrastructure. Discovery Collector extends the Discovery process beyond archived data, and enables any ESI identified to be securely archived and retained within Enterprise Vault, making the content available for review by Discovery Accelerator.

The Discovery Collector services can potentially be resource-intensive, and may require high capacity storage to retain source data and indexes.

In most situations, you should not co-host any other virtual machines with Discovery Collector. The Discovery Collector server may need to be scaled out, depending on the expected use.

Discovery Collector needs to be appropriately planned and sized. See the Discovery Collector *Installation and Planning guide* for full planning and sizing guidance. The following information provides a high-level overview of requirements but should not be used for sizing purposes.

Hardware requirements

The Discovery Collector infrastructure consists of VMware virtual machine image nodes that can be scaled to meet the environment requirements. There are three different node types defined: Controller, Full-text, and Compute.

Node type	Description
Controller	VM within the cluster, one required per cluster, where Web application and the core application databases are managed.
Full-text	VM within the cluster that manages the full-text indexes. One required per cluster.
Compute	<p>Nodes within the cluster that do all of the harvesting, indexing, and data migration, including desktop collections. The number of compute nodes will vary within a Discovery Collector system as a function of the data to be indexed and managed. At least one compute node is required per cluster.</p> <p>The processing load is symmetrically distributed across all compute nodes. The first of the compute nodes is called the “action node”; it coordinates work load across the compute nodes. The indexes that are created during harvesting and data migration (when applicable) are stored/cached within the compute nodes until an index “load operation” is executed. At that time the cached indexes are merged into the NAS databases and managed by the Controller and Full-text node.</p>

Discovery Collector can be configured as a standalone system or scaled out as a clustered system. The table below illustrates various potential configurations:

Discovery Collector system configuration	Maximum managed terabytes	Physical ESX servers	Virtual machine count
Standalone system	1	1	1
Single ESX Server	3	1	3
2 ESX Servers	10	2	4
3 ESX Servers	30	3	6
4 ESX Servers	60	4	8
5 ESX Servers	90	5	10

Multiple Discovery Collector systems can be associated and managed through the use of a Federation System Configuration. See the Discovery Collector *Installation and Planning guide*.

Due to the potentially large volumes of data transferred, it is recommended that Discovery Collector has access to the enterprise infrastructure through gigabit network technology.

Server considerations

Each Discovery Collector ESX server should have a dedicated x64-based server with at least four processors available. Either four physical CPUs or a combination of multi-core and multi-CPU can be used, but server sizing must not be based upon hyper-threading. See the Discovery Collector *Installation and Planning guide* for more information.

A recommended ESX server should consist of the following:

- Minimum number of CPU cores: 4.
- Memory: 32 GB for a multiple ESX server configuration; 48 GB for a single.
- Storage: 300 GB or 450 GB, depending on the requirements of the installation. Each cluster node requires 150 GB for its virtual disk.

The Discovery Collector servers may need to be scaled out, as briefly discussed in the following sections.

Storage considerations

Discovery Collector requires storage for VMware virtual disks, metadata indexes, full-text indexes, and temporary caches.

- All Discovery Collector installations require virtual disk storage for each installed virtual machine.
- A cluster installation requires additional storage supplied by a NAS system for its metadata indexes, full-text indexes, and temporary caches. NAS systems provide scalable storage to satisfy the requirements for cluster installation managing from 3–90 TB on data sources requiring 1.6–25.6 TB of Discovery Collector storage.
- A standalone installation requires an additional virtual disk for its metadata indexes, full-text indexes, and temporary caches. The additional storage requirements to manage up to 1 TB on data sources requires an additional 250 GB virtual disk.

Discovery Collector tuning

Install and tune Discovery Collector as detailed in the Discovery Collector *Installation and Planning guide*. Particular attention should be paid to tuning the harvester processes.

Desktop collection

Desktop collection can potentially impact the performance of the target machines depending upon their specification. If possible, desktop collection should be scheduled out of normal business hours.

Desktop collection jobs should be configured to process five or six desktops in parallel.

Architecture considerations

Discovery Collector needs to be appropriately planned and sized. See the Discovery Collector *Installation and Planning guide* for full planning and sizing guidance.

Medium and large customers may wish to consider the addition of a new Enterprise Vault server dedicated to archiving data gathered by Discovery Collector.

If Discovery Collector is required in more than one site, consider whether a federated system is appropriate. This allows you to see all the Discovery Collector systems from a single interface, and allows search across multiple systems at once.

Monitoring Discovery Collector

Monitor the Discovery Collector infrastructure during the following activities to ensure that the environment is performing correctly:

- During harvest processing (collecting and indexing).
- Executing a policy to move or copy to retention server (Enterprise Vault).
- Executing a Discovery export policy to copy, format and deduplicate the data from Discovery Accelerator to retention server (file server) in load file format.

The Discovery Collector dashboard reports the current throughput metrics, and VMware Virtual Center can be used to monitor hardware resource utilization.

Enterprise Vault infrastructure

The introduction of Discovery Accelerator can have a significant impact on the existing Enterprise Vault architecture. The Enterprise Vault performance also influences Discovery Accelerator considerably.

If the Enterprise Vault environment is being sized along with the Discovery Accelerator implementation, the sizing can take into account the impact of Discovery Accelerator. However, it is likely that you are adding Discovery Accelerator to an existing Enterprise Vault environment, and therefore you may need to upgrade the existing servers or add further ones.

Medium and large customers may also want to consider the addition of a new Enterprise Vault server specifically for archiving data gathered by Discovery Collector.

Enterprise Vault indexing service considerations

Discovery Accelerator search functionality can have a significant effect on the Enterprise Vault indexing services. Tuning these services can greatly improve Discovery Accelerator search efficiency and time to complete.

Enterprise Vault 10.0 introduced a 64-bit indexing engine and continues to support any existing 32-bit indexes. In addition, Enterprise Vault 10.0 also introduced index groups to provide a distributed and scalable indexing architecture. Index groups can offer a method of scaling-out and distributing index volumes to achieve a greater search throughput.

Discovery Accelerator 10.0 supports searching 32-bit indexes that are available in Enterprise Vault 9.0 and 10.0, and 64-bit indexes that are available in Enterprise Vault 10.0.

Server considerations

You may need to upgrade an existing environment to meet the higher specifications that the Discovery Accelerator services demand. In addition, with Enterprise Vault 10.0 you may want to consider the use of index groups to provide dedicated index servers. See the Enterprise Vault 10.0 *Indexing design and implementation best practice guide* at <http://www.symantec.com/docs/DOC4250>.

The servers that are hosting the Enterprise Vault indexing service may need to handle sustained periods of CPU, memory, and I/O-intensive activity during searches with, by default, ten concurrent searches running until all indexes have been searched. This activity is likely to be repeated throughout the working day as the legal discovery team adds search requests.

This activity needs to run alongside any normal journaling, archiving, retrieval, or end-user loads. A multi-processor server is essential to ensure that all these concurrent activities can function with reasonable performance. The following table recommends the minimum number of processor cores needed per indexing server.

	Enterprise Vault 9.0	Enterprise Vault 10.0
Small customer Enterprise Vault server	2 cores	4 cores
Small customer Enterprise Vault with Discovery Accelerator	4 cores	8 cores
Medium/large customer Enterprise Vault server	4 cores	8 cores

Either physical CPUs or a combination of multi-core and multi-CPU can be used, but server sizing must not be based upon hyper-threading.

The Enterprise Vault indexing services can require a high memory capacity, particularly when complex searches may be submitted to multiple indexes concurrently.

A 32-bit Windows server running an Enterprise Vault 9.0 indexing service should have a minimum of 4 GB of RAM and can have up to 64 GB of RAM installed, depending on Windows edition, using the /PAE boot flag. Additional RAM can offer significant performance benefits by providing additional physical memory beyond 4 GB for use between multiple concurrent 32-bit index server processes, reducing overheads such as disk paging between concurrent processes.

Note: You should not use the /3GB boot flag to attempt to extend the per-process limit to 3 GB. This will not increase the memory available to each index process and has the potential to cause issues with memory, file handling, and network connectivity.

In addition, the Enterprise Vault 9.0 services can be installed in an x64 environment, which can offer further benefits in efficiency such as additional memory for the Windows system file cache, and can improve index server performance by reducing I/O overheads. See “Storage considerations” below for more information.

Enterprise Vault 10.0 introduces a 64-bit indexing engine and also supports existing 32-bit index volumes. Enterprise Vault 10.0 must be installed on Windows 2008 R2 x64.

The following table recommends the minimum memory that should be provisioned:

	Enterprise Vault 9.0	Enterprise Vault 10.0
Small customer stand-alone Enterprise Vault	4 GB *	8 GB
Small customer single server Enterprise Vault and Discovery Accelerator	8 GB	16 GB
Medium/large customer	12 GB – 16 GB	16 GB

* If using 64-bit Windows, the minimum memory should be 8 GB

Storage considerations

Typical Discovery Accelerator search loads require the index files to be available on high-speed storage. An Enterprise Vault server also uses other storage devices not discussed in this section for purposes such as the Enterprise Vault storage service. The type of storage and interfaces used must ensure that the indexing storage does not become a significant bottleneck.

Enterprise Vault 10.0 indexing services use the EV server cache location to temporarily store result metadata, which can result in high I/O loads. The Enterprise Vault server cache should be stored on dedicated high-speed storage, and it should not be co-located with any other Enterprise Vault storage such as indexes and vault stores.

Discovery Accelerator needs to retrieve all item metadata that matches a specified query in each index volume. It retrieves these items from an index volume in batches of 50,000 items by iteratively querying the index. Each search iteration passes through several phases of activity but, at a high level, it first identifies the hits for the required batch and then retrieves all the result metadata for the desired quantity of hits.

The I/O load and duration of searches depends on a number of factors, which include the following:

- Enterprise Vault server processor and operating system memory architecture.
- Enterprise Vault server RAM availability.
- Enterprise Vault server CPU contention.
- Enterprise Vault server Windows system file cache size/state and external file caches (for example, SAN cache).
- Available 32-bit index server processes at each indexing service [where relevant].
- 64-bit indexing engine capacity (closely related to available RAM).
- I/O bandwidth.
- Search complexity.
- Total number of hits per index searched.
- Concurrent search requests including end-users and other Discovery search threads. (Potentially contenting for memory and I/O.)
- Distribution of index volumes across Enterprise Vault indexing services

Enterprise Vault 10.0 provides index groups to help dedicate resources and scale indexing to meet demands.

It is likely that the available IOPS will be shared between ten concurrent search threads per server. On a typical Enterprise Vault 9.0 32-bit server, this is unlikely to provide the full potential throughput due to contention between the threads at the storage increasing latency. The use of Windows x64 and additional RAM can help mitigate the issues. The number of 32-bit index server processes and search threads may also be tuned according to “Search tuning” on page 51.

Note: The use of Windows x64 with Enterprise Vault 9.0 enables the system file cache to grow and improve the I/O to 32-bit index files. This can lead to a reduction in the 32-bit index I/O load to irregular peaks of possibly several hundred IOPS, with little physical I/O for the duration of the result metadata retrieval. Installing each Enterprise Vault 9.0 server on a Windows x64 system with at least 12 GB to 16 GB of RAM should significantly improve performance and reduce the I/O load.

The result metadata is passed back to Discovery Accelerator, which in turn adds the metadata to the Discovery customer database. The Database server then needs to further process the data from all concurrent searches, and it is possible that the collective Enterprise Vault servers could produce the results faster than they can be processed at the database. Therefore, no individual Enterprise Vault server needs to provide results at a rate greater than 2,000 per second.

Sizing the indexing I/O subsystem

Each Enterprise Vault indexing service storage should be independently sized to achieve the desired overall throughput. By default, Discovery Accelerator searches ten index volumes concurrently per indexing service, so any latency or contention at the storage significantly affects all search times at that indexing service. This is in addition to any index updates that may need to be done during normal archiving activities.

Index volume search rate

A search with the default of ten threads typically demands 600 IOPS for 32-bit index volumes or 1,200 IOPS for 64-bit index volumes whilst searching. (However, this varies depending on the hits retrieved per volume.) It should search approximately 5,000 – 6,000 mailbox index volumes per hour excluding result retrieval, which will reduce this rate.

Increasing the number of search threads to raise this iteration rate also increases the I/O load and memory footprint, and it may not achieve any improvement if the index locations are not suitably sized or there is not sufficient memory.

The optimum number of active index servers varies between environments and should be determined through tuning. See "Search tuning" on page 51 for more information.

Result retrieval

Small customers may want to size each Enterprise Vault server to retrieve up to 500 results per second, and medium or large customers should size each Enterprise Vault server to retrieve up to 2,000 results per second.

For Enterprise Vault 9.0, small customers should provision storage which can deliver around 600 IOPS. For medium or large customers, the best approach to meeting these performance demands would be to provision an array delivering 1,000 IOPS and use Windows x64 with 12 GB to 16 GB of RAM on each index server.

For Enterprise Vault 10.0, small customers should provision storage which can deliver around 1200 IOPS. For medium or large customers, the best approach to meeting these performance demands would be to provision an array delivering 3,200 IOPS on each index server.

The following high-level rules of thumb can be applied to select the most appropriate storage for each indexing service:

	32-bit index volumes only	64-bit or mixed 32/64-bit index volumes
Small customer stand-alone Enterprise Vault	500 IOPS	1200 IOPS
Small customer single server Enterprise Vault and Discovery Accelerator	500 IOPS	1200 IOPS
Medium/large customer	1000 IOPS	3200 IOPS

Consider the example of a simple search of a single journal archive containing 22,000,000 items and the search hits 300,000 items:

- 32-bit index volumes: the archive would typically consist of 74 index volumes, and can result in a sustained load varying between 600 and 2,000 IOPS for 17 minutes.
- 64-bit index volumes: the archive would typically consist of five index volumes, and can result in a sustained load around 3,200 IOPS for 7 minutes.

The indexing service file locations require good random access or a high number of IOPS combined with good transfer rates. Therefore, a striped array of many disks should be used (using hardware-based RAID rather than software-based RAID).

Sharing a storage device between multiple indexing services can cause severe performance degradation.

When you select storage, consider the advice in "Storage" on page 10, and create and align partitions as advised.

An Enterprise Vault indexing server should typically have the following partitions:

- System drive (RAID-1)
- Index locations (RAID-5 or RAID-10 striped arrays of many high-speed disks)
- EV 10.0 Server Cache (redundant striped array of several high-speed disks)

Network-attached storage (NAS)

You may have already chosen network-based storage for index locations before you required Discovery Accelerator. In some instances, you may have used a single NAS device as a central file location for multiple or all indexing services.

The performance of NAS storage may be sufficient for basic indexing purposes to be used by lightweight applications such as the Enterprise Vault Web/desktop search

application, which conducts a simple search and retrieves ten results only. However, depending on the specification, this may present a significant bottleneck for Discovery Accelerator searches.

Sharing a storage device between multiple indexing services can cause severe performance degradation.

If an existing environment uses a NAS device to store the indexes, it may need to be reconsidered and indexes moved to a faster location.

A NAS device should not be used for the Enterprise Vault server cache.

For Enterprise Vault 10.0, see the *Indexing design and implementation best practice guide* (<http://www.symantec.com/docs/DOC4250>).

32-bit and 64-bit index volumes

Enterprise Vault 10.0 introduced 64-bit index volumes. In upgraded environments, a new 64-bit index volume is created for each existing archive, and potentially Discovery Accelerator may need to search a large number of new index volumes.

To take advantage of the features and performance offered by the new 64-bit indexing engine, it would be beneficial to upgrade search target 32-bit index volumes to 64-bit. See the *Indexing design and implementation best practice guide* (<http://www.symantec.com/docs/DOC4250>).

Index detail level

The Enterprise Vault indexing service offers several levels of indexing detail.

Indexing levels for Enterprise Vault indexing service

Level	Notes
Brief	The index created by Enterprise Vault enables searches on the following attributes of each item: Author, Subject, Recipient, Created Date, Expiry Date, File Extension, Retention Category, and Original Location. If a search matches an attachment to an item, the search result contains both the main item and the attachment.
Medium (32-bit only)	As for Brief, and in addition enables searches on the content of each item, excluding phrase searches.
Full	As for Brief, and in addition enables searches on the content of each item, including phrase searches.

You can specify the default indexing detail level for archive indexes in various places. Once items have been stored in a particular archive and the index has been created, you cannot change the index detail level without rebuilding that index.

To take advantage of Discovery Accelerator's content phrase-based searching, ensure that all the indexes that will be searched use the Full level. We strongly recommend that you set the index detail level for all archives to Full.

64-bit index volume tuning

If an Enterprise Vault 10.0 indexing service is hosted on a server that is less than the recommended specification, and it is not a dedicated indexing server, then the server could become overloaded.

The following Enterprise Vault Server extended setting can be modified to reduce contention on resources. This setting should be tuned depending upon server specification and workload through experimentation to ensure an indexing backlog does not start to form.

Setting	Default	Recommended
Maximum concurrent indexing capacity	30	5 – 30 depending on resource availability.

32-bit index volume tuning

Index granularity

Each 32-bit index is created using a schema that defines how the index should be structured. Part of the schema is the document granularity, which defines whether items with attachments are stored separately (attachment granularity) or merged together as a single item (item granularity). Once a 32-bit index has been created with its schema, it cannot be changed without rebuilding the index.

The default schema depends on the type of archive and version of Enterprise Vault installed when the index was first created. All Enterprise Vault versions prior to 8.0 created all indexes as attachment granularity. At 8.0, all new journal archives are now created as item granularity, and this was subsequently reflected in Enterprise Vault 2007 SP5 for all new journal archives.

When attachment granularity indexes are used, Discovery Accelerator searches that include content or subject terms will result in two searches being started internally on each index: one searching the top-level items only, and the other searching top-level items and attachments. This generates two result sets that have to be joined in memory at the indexing service. This extra work can quickly consume both memory and CPU resources at the Enterprise Vault indexing service, slowing the search and potentially running out of resources.

In the item granularity schema, when an item and its attachments are merged together, all the fields of the top-level item and its attachments are merged together as a single item. The main advantage to item granularity is that only a single internal search of the top-level items is now required, significantly reducing the processing required during searching.

Note: Merging all fields from the top-level item and its attachments into a single item may result in some unexpected search results, in particular with AND terms. Searching for two terms can return a hit even if each term is located in different attachments or fields, as they are all now considered a single item. For example, searching for “Meeting” in the content AND a document type of “HTML” may return a hit where “Meeting” was found in an attached text document but another attachment happens to be an HTML document. This impact must be discussed with the legal discovery team.

Changing the document granularity on a 32-bit index also impacts how search results are displayed in the Enterprise Vault search application for that index. Attachment granularity displays the exact document (whether an attachment or top-level message) that matches the search criteria. With item granularity, even if the matched document is an attachment, the search application only displays the related top-level item. (However, this item does contain links to the attachments, and the user can easily look through each attachment to locate the match.) The original item can also be viewed (which involves a retrieval from storage) to see the attachments. This does not affect how Discovery Accelerator displays items in review.

You need to consider the impact to the functionality when using item granularity against the potentially significant performance improvements.

A compromise could be to change all journal archives to item granularity while leaving mailbox vaults, which are usually used by end-users, as attachment granularity. All 32-bit journal index volumes created post Enterprise Vault 8.0 are created as item granularity.

Note the following:

- All new 64-bit index volumes available in Enterprise Vault 10.0 will produce results the same as item granularity.
- Changing the document granularity for existing 32-bit index volumes requires the volumes to be rebuilt, which may be time-consuming. In addition, Enterprise Vault 10.0 will not rebuild 32-bit index volumes but instead will upgrade to 64-bit index volumes.

Moving to Enterprise Vault 10.0 and 64-bit index volumes will require rebuilding the existing archives. Therefore, you should consider whether it is worth the time and costs in rebuilding 32-bit index volumes with Enterprise Vault 9.0, or would it be worth upgrading to Enterprise Vault 10.0 and 64-bit index volumes to gain their benefits.

Add or modify the following registry key to each Enterprise Vault indexing service. The key applies to all indexes on the server.

Location: HKLM\SOFTWARE\KVS\Enterprise Vault\Indexing\

Or, on Windows x64:

Location: HKLM\SOFTWARE\Wow6432Node\KVS\Enterprise Vault\Indexing\

Key: SchemaType (DWORD) set to 1

Note: To set on a per-index basis, create a sub-key of \Indexing named using the specific archive “vault entry id”.

Maximum locations per index

The 32-bit indexing service uses internal structures known as “locations” to determine when to split an index into multiple index volumes. These locations roughly equate to the total number of text words within all successfully converted documents and the document attributes that have been indexed.

The default depends on the version of Enterprise Vault installed when the index was first created.

- All Enterprise Vault versions prior to 8.0 allow just over four billion locations per index when creating new index volumes. Each index volume is approximately 19 GB in size, which represents about five million typical email messages.
- At Enterprise Vault 8.0, all new index volumes default to one billion locations, which represents about 350,000 typical email messages or 60,000 typical files.

However, you can change this default on an indexing service or per-vault basis.

Location: HKLM\SOFTWARE\KVS\Enterprise Vault\Indexing\

Or, on Windows x64:

Location: HKLM\SOFTWARE\Wow6432Node\KVS\Enterprise Vault\Indexing\

Key: AVSMaxLoc (DWORD) decimal value between 500,000,000 and 4,294,000,000

Note: This applies to all newly created 32-bit index volumes. To set on a per-index basis, create a sub-key of \Indexing named using the specific archive vault entry id. However, changing the number of locations for existing vaults requires the indexes to be rebuilt, which may be time-consuming. Enterprise Vault 10.0 will not rebuild 32-bit index volumes but instead will upgrade to 64-bit index volumes.

Reducing the maximum number of structural locations per index reduces the size of the index volumes. However, it also increases the total number of index volumes, and so may slightly increase the overall capacity requirements and create more index volumes to search. Creating more index volumes may slow Discovery Accelerator searches by requiring them to iterate through more indexes, but this can be countered by the benefits below.

Reducing the size of the indexes can provide the following benefits:

- Enables more efficient searching when date ranges are used.
- Reduces the memory requirements of searching the archive with complex terms.
- Enables smaller sections of an archive's index to be rebuilt if required, reducing rebuild time.
- Reduces the risk of index corruption impacting service.
- Potentially reduces the I/O load during Discovery searches, improving search times.

The default was set to 1,000,000,000 to achieve a balance between index size and number of index volumes.

Index file locations

The Enterprise Vault indexing service distributes its index files between the file locations specified to the service in a round robin fashion. This may help to distribute the I/O load during Discovery Accelerator searches and provide better performance. Once an index has been created, it is not moved between the locations, so adding locations to an existing system may not immediately bring about any improvements until new index volumes are added.

Locate each file location on a separate disk array to prevent contention between concurrent searches of different indexes. A minimum of four different index file locations is generally suggested, ensuring that each location is on a different disk array.

Avoid LAN-based storage for index file locations because of the I/O load. It is particularly important that index file locations from multiple Enterprise Vault indexing services should not be located on the same disk spindles, if possible. This can create a storage bottleneck and negatively affect search performance.

Index file 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. Any scheduled defragmentation should be performed with the indexing service stopped to prevent the potential for corruption.

Note: See the Enterprise Vault *Compatibility Charts* for certified automated background defragmentation tools.

System tuning

On 32-bit Windows running Enterprise Vault 9.0, do not use the operating system boot flag /3GB. This does not provide any benefit and can result in running out of system page table entries.

Disable Windows file indexing on the drives that contain Enterprise Vault indexes.

If you have installed anti-virus software to scan file and network accesses, disable it on the index servers. Anti-virus software should exclude the index locations and the Enterprise Vault server cache from its file scan due to potential issues with anti-virus software corrupting indexes.

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 and in Windows.

Enterprise Vault storage service considerations

The Discovery Accelerator pre-fetch cache, export feature, online reviewing, and analytics data collection can create an unexpected load at the Enterprise Vault storage services.

Server considerations

You may need to upgrade an existing environment to meet the high demands of the Discovery Accelerator services.

The server that is hosting the Enterprise Vault storage service may need to handle sustained periods of CPU, memory, and I/O-intensive activity during exports and analytics original item retrieval.

By default, each Discovery Accelerator export starts 25 concurrent retrieval threads, up to a maximum of 100 threads, which run until all the items have been downloaded. This activity is likely to be repeated throughout the working day as the legal discovery team initiate export requests.

Each Discovery case that is enabled for analytics needs to download all the associated messages. By default, it starts two concurrent retrieval threads per Enterprise Vault server, although this might be increased to improve collection. These threads run until all the items have been downloaded—by default, up to 10 threads across all Enterprise Vault servers per concurrently enabled case.

The Discovery Accelerator case review features can also generate on-demand downloads from the Enterprise Vault storage service. The Discovery Accelerator pre-fetch cache discussed previously can help minimize the potential impact of online reviewers by downloading the most recently searched items during a predefined time window.

All this activity needs to run alongside any normal Enterprise Vault journaling, archiving or end-user tasks. A multi-processor server is essential to ensure that all these concurrent activities can function with reasonable performance. The following table recommends the minimum number of processor cores needed per indexing server.

	Enterprise Vault 9.0	Enterprise Vault 10.0
Small customer Enterprise Vault server	2 cores	4 cores
Small customer Enterprise Vault with Discovery Accelerator	4 cores	8 cores
Medium/Large customer Enterprise Vault server	4 cores	8 cores

Either physical CPUs or a combination of multi-core and multi-CPU's can be used, but server sizing must not be based upon hyper-threading.

The Enterprise Vault 9.0 services are 32-bit, so each index volume process is limited to 2 GB of virtual memory. A 32-bit Windows server running Enterprise Vault 9.0 services should have a minimum of 4 GB of RAM and can have up to 64 GB of RAM installed using the /PAE boot flag. Additional RAM can offer significant performance benefits by providing additional physical memory beyond 4 GB for use between multiple concurrent processes, reducing overheads such as disk paging between concurrent processes. This can be of particular benefit to indexing; see "Enterprise Vault indexing service considerations" on page 65.

Note: You should not use the /3GB boot flag to attempt to extend the per-process limit to 3 GB. This will not increase the memory available to each index process and has the potential to cause issues with memory, file handling, and network connectivity.

In addition, the Enterprise Vault 9.0 services can be installed in an x64 environment, which can offer further benefits in efficiency such as additional memory for the Windows system file cache, and can improve index server performance by reducing I/O overheads. See "Storage considerations" below for more information.

Enterprise Vault 10 introduces a 64-bit indexing engine whilst supporting any existing 32-bit index volumes. Enterprise Vault 10 must be installed on Windows 2008 R2 x64.

The following table recommends the minimum memory that should be provisioned:

	Enterprise Vault 9.0	Enterprise Vault 10.0
Small customer stand-alone Enterprise Vault	4 GB *	8 GB
Small customer single server Enterprise Vault and Discovery Accelerator	8 GB	16 GB
Medium/large customer	12 GB – 16 GB	16 GB

* If using 64-bit Windows, the minimum memory should be 8 GB.

Storage considerations

Under normal circumstances, the archived data that the Enterprise Vault storage services manage is likely to be stored on slower, archive-level storage or off-line storage devices. This is likely to continue to be acceptable and preferable for the majority of cases.

However, larger customers who demand high-speed responses may want to store archived data on higher specification devices to ensure that online reviewing,

analytics, and export activities are not delayed. Many devices that are targeted at archive data or for compliance are already capable of providing a high throughput of retrievals, and so should remain suitable for medium and large customers.

When Discovery Accelerator online reviewing, analytics data collection, and exports are running, the Enterprise Vault storage services are queried to obtain the original archived data. This creates a higher I/O load than normal, which may cause contention with normal journal, archiving, and end-user tasks. If multiple Enterprise Vault servers share a single storage device for all vault stores, the load generated (particularly by analytics data collection) can result in a significant bottleneck at the storage device.

Ensuring that the Discovery Accelerator pre-fetch cache is operational at an appropriate time window should reduce activity at undesirable times. The Discovery Accelerator pre-fetch cache downloads original data at out-of-hours times.

Scaling Enterprise Vault for Discovery Accelerator

The Enterprise Vault architecture is likely to need scaling up to meet the demands of Discovery Accelerator and the archiving load expected. However, scaling up individual Enterprise Vault servers does have limited benefit. Therefore, to meet service-level expectations, the Enterprise Vault infrastructure may need scaling out by adding additional Enterprise Vault indexing and storage servers.

Scaling out

Each Enterprise Vault service can be hosted on different physical servers, but distributing every service on a separate server does not necessarily provide any benefit.

With Enterprise Vault 9.0, it is recommended that the Enterprise Vault indexing and storage services should always be co-located. The indexing service should handle the indexes for archives that are stored by the co-located storage service. To scale out the Enterprise Vault indexing and storage services, add additional servers with both services and distribute the archives over these servers, potentially using move archive.

Enterprise Vault 10.0 introduces index groups, which enable a set of dedicated and scalable servers to manage a set of index volumes. See the *Enterprise Vault 10 Indexing design and implementation best practice guide* (<http://www.symantec.com/docs/DOC4250>).

The addition of more storage services and vault stores should not affect the SIS ratio because of the enhanced sharing-group single instance storage.

When you use an EMC Centera device, the SIS is handled differently than NTFS partitions or other network-based storage, and it is shared across vault stores located on the same Centera. Therefore, the addition of servers and vault stores to the same Centera does not affect the SIS.

Rolling over journal vaults

The journal archives are likely to grow quickly, resulting in very large archives with large indexes. To keep journal archives and their indexes within manageable sizes, you can roll over the journal archives on a regular basis (for example, monthly). Rolling over journal indexes also provides more control as to where the data and indexes are located. To prevent the Enterprise Vault servers from becoming oversubscribed with search requests, you can add further Enterprise Vault servers and add all new journal archives to them. This effectively makes the previous Enterprise Vault servers read-only servers, improving the search performance on them.

Monitoring Enterprise Vault

The Enterprise Vault documentation recommends methods for monitoring the Enterprise Vault environment. You should also monitor the indexing and storage services to measure the impact of Discovery Accelerator as follows:

- The impact of searches to allow the Discovery search threads to be correctly tuned.
- The impact of the pre-fetch cache or production upon the storage service.
- The impact of Discovery online reviewing, if the pre-fetch cache is not used.

Use Windows Performance Monitor to obtain system statistics. You should typically monitor the following counters:

Object	Counters	Instances
LogicalDisk	Avg. Disk Queue Length	Index location drives
	Disk Transfers/sec	
	Avg. Disk Bytes/Transfer	
	Avg. Disk sec/Transfer	
	Disk Bytes/sec	
	Split IO/sec	
Server	Files Open	

Object	Counters	Instances
Memory	Page Faults/sec	
	Pages/sec	
	Available Bytes	
	Free System Page Table Entries	
Network Interface	Output Queue Length	All instances
	Packets/sec	
	Bytes Total/sec	
Processor	% Processor Time	_Total
Process	% Processor Time	All Indexserver.exe processes
System	Processor Queue Length	
	Context Switches/sec	
	File Data Operations/sec	
	Processes	

CPU and memory

The % Processor Time for the _Total counter indicates overall system activity. If the CPU is consistently 80%, the server may start to indicate performance issues.

However, consistently high CPU is not the only indicator of performance problems. If the index server processes tend to use very little CPU for long durations, this could indicate an I/O bottleneck. Monitoring the index server processes for low CPU is useful when the index file locations are not located on the index server itself, but on a remote storage server.

It would be worth monitoring the memory counter Pages/sec, which records the number of hard faults to disk. If there are significant or sustained hard faults, the problem should be traced to the source.

Disk

The disk counters are only relevant if the index file locations are locally attached. If the index file locations are on a remote server, monitor that server for disk activity. Typically, the disk read/write queue length counters are monitored for evidence of a disk bottleneck. The queues should not normally rise above twice the number of disks in the array.

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

The use of these counters may not be appropriate when using a SAN. Use the hardware vendor's tools instead.

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

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

The Server counter Files Open can also be useful to indicate how many files are in use at any time, perhaps suggesting that the server is oversubscribed with search requests.

End-user advice

The methods that end-users employ can influence the overall Discovery Accelerator system performance. We recommend that you distribute the following advice to users of Discovery Accelerator.

Legal holds

Legal holds can affect the performance of search acceptance. Therefore, it is best to create a case initially without legal holds and then, once the bulk of searches have been accepted, enable legal holds on the case.

Searching

Searching can potentially be time-consuming, but you can improve performance by ensuring that your searches are appropriately created and focused. A poorly constructed search can inadvertently cause thousands of indexes to be searched for large volumes of data, perhaps unnecessarily. Following these simple practices helps to improve overall search performance.

- Ensure that the case is searching only the archives that are relevant to the case in the case properties.
- Do not start a search without criteria. This retrieves every item from all indexes.
- Do not use wildcards unless necessary. This can severely impact performance.
- Make searches specific, and try to include author or recipient details.
- Specify date ranges. This can reduce the number of indexes searched.
- Avoid overusing search terms. Thousands of terms can cause iterative searches.
- Ensure that scheduled searches do not run at the same time as system backups.
- Quickly accept or reject searches to avoid filling and slowing the database.
- Test new searches in research folders, and then delete the folders as necessary.

Analytics

When you enable a case for analytics, the original items are downloaded from Enterprise Vault into Discovery Accelerator. This can place a high load on the Discovery services, but you can improve performance by following some simple practices:

- Try to avoid enabling a case for analytics during office hours. If possible, enable the case at the end of the day to retrieve message content overnight.
- Avoid enabling several cases for analytics simultaneously. Enable the cases one at a time and wait for the previous case to be marked as completed before you enable the next.
- Define rules for automatic categorization before you enable a case for analytics.

Reviewing

The online reviewing facility provides an integrated, feature-rich environment with which you can review large volumes of documents before you produce them. An appreciation of the processes triggered during reviewing can prevent unnecessary load on the service.

The message preview pane may initially take several seconds to display. This is expected. During this time, the service fetches the archived item from Enterprise Vault, which may be located on slower archive equipment. Do not attempt to speed up the display by clicking several different items.

Export and production

When you export or produce search results, Discovery Accelerator starts to download the original data from Enterprise Vault to the desired export location. If you select a poor destination, such as a location over the network, the export can take significantly longer.

The Discovery Accelerator administrator should provide several different high-speed locations on the Discovery Accelerator server in which to place exports.

- Export to a high-speed drive located on the Discovery Accelerator server.
- Try to avoid running simultaneous exports. If this happens, try to ensure that they are output to different drives.
- Exporting Exchange email to native format is much faster than to PST, and it enables easier tracking and authentication through the production reports.

- If an export fails for any reason, use the export retry feature rather than re-export.
- Only export to HTML if necessary.

Note: You do not need to break a large export into several smaller exports, as required in previous versions. However, splitting a large export into several smaller exports may help you to administer larger volumes of data.
