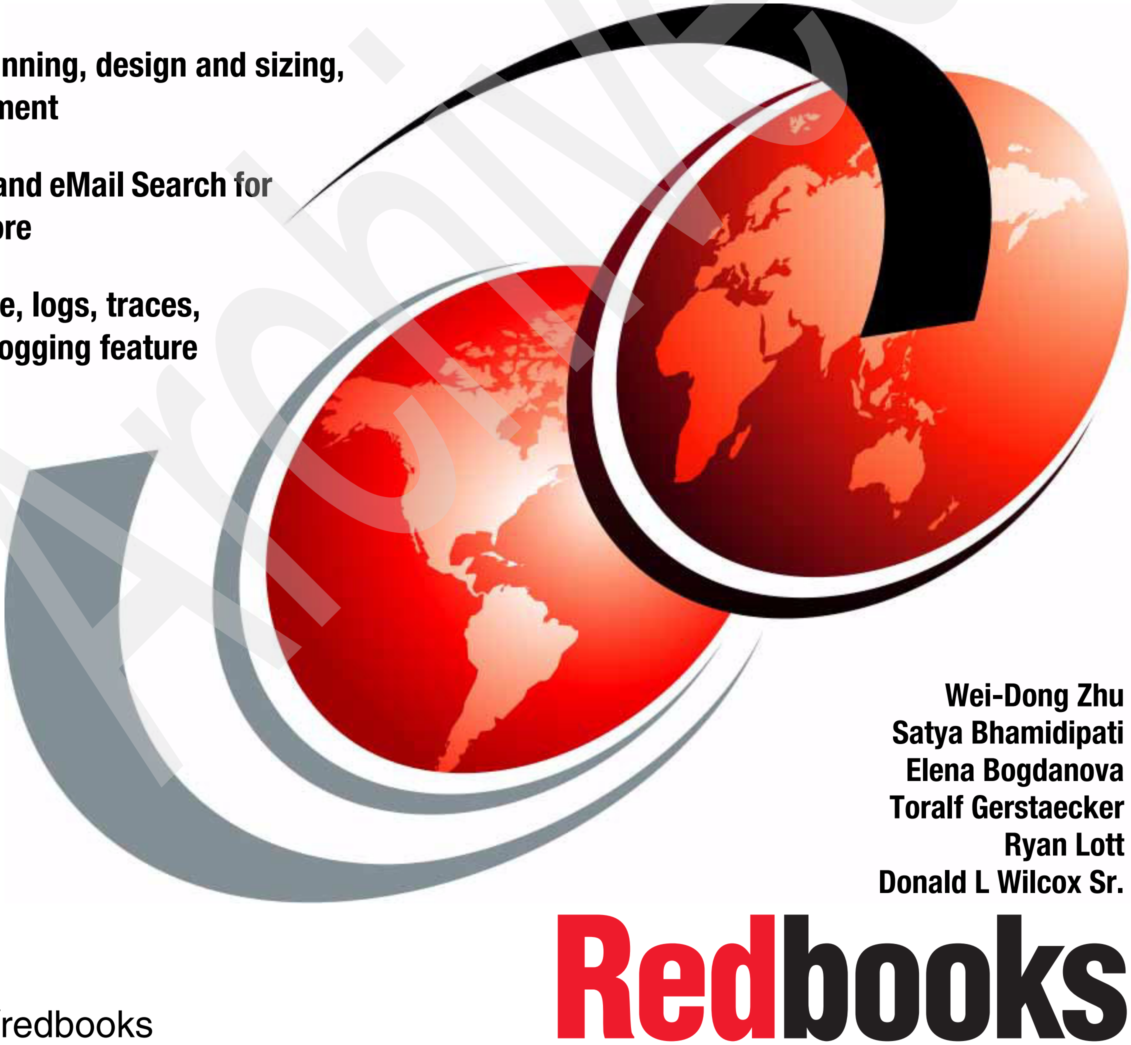
Best Practices for Setting Up an IBM CommonStore Solution

For Mailbox Management, e-Mail Retention and Discovery



- Solution planning, design and sizing, and deployment
- Journaling and eMail Search for CommonStore
- Maintenance, logs, traces, and report logging feature

Wei-Dong Zhu
Satya Bhamidipati
Elena Bogdanova
Toralf Gerstaecker
Ryan Lott
Donald L Wilcox Sr.

Redbooks

ibm.com/redbooks

International Technical Support Organization

Best Practices for Setting Up an IBM CommonStore Solution: For Mailbox Management, e-Mail Retention and Discovery

December 2006

SG24-7325-00

Note: Before using this information and the product it supports, read the information in "Notices" on page xi.

First Edition (December 2006)

This edition applies to Version 8, Release 3, Modification 1 of IBM CommonStore for Exchange Server (program number 5724-B85), Version 8, Release 3, Modification 1 of IBM CommonStore for Lotus Domino (program number 5724-B86), and IBM eMail Search for CommonStore V8.

Contents

Figures

Tables

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:
IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785 U.S.A.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

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

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

Trademarks

The following terms are trademarks of the International Business Machines Corporation in the United States, other countries, or both:

AIX®
DB2 Universal Database™
DB2®
Domino.Doc®
Domino®
eServer™
IBM®
iNotes™
iSeries™
Lotus Notes®
Lotus®
MVS™
Notes®
OS/400®
pSeries®
Redbooks (logo) ™
Redbooks™
Sametime®
System p5™
Tivoli®
WebSphere®
Workplace™
xSeries®
z/OS®

The following terms are trademarks of other companies:

Oracle, JD Edwards, PeopleSoft, and Siebel are registered trademarks of Oracle Corporation and/or its affiliates.

Siebel are registered trademarks of Oracle Corporation and/or its affiliates.

SAP, and SAP logos are trademarks or registered trademarks of SAP AG in Germany and in several other countries.

Java, Solaris, Sun, and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Active Directory, Excel, Internet Explorer, Microsoft, Outlook, PowerPoint, SharePoint, Windows NT, Windows, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Pentium, Intel logo, Intel Inside logo, and Intel Centrino logo are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States, other countries, or both.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

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

Preface

This IBM® Redbook helps you to set up CommonStore solutions for mailbox management or long-term e-mail retention and discovery purposes. We provide an overview of the key products and components that comprise the solutions:

- IBM CommonStore for Lotus® Domino® V8.3
- IBM CommonStore for Exchange Server V8.3
- IBM eMail Search for CommonStore V8.3

> **Important:** Use this IBM Redbook in conjunction with the CommonStore product manuals and eMail Search documentation. We only address the specific topics that we believe will help you better in understanding the solutions, and preparing and setting up CommonStore solutions.

We cover the following topics:

- *Solution planning*, where we provide some insight to help clients, project managers, sales people, and technical staff to avoid common pitfalls during the planning and setup phase of an implementation project of a CommonStore solution.
- *Solution design*, where we address the key areas to consider when planning and designing a solution including a set of requirement-gathering questionnaires.
- *Sizing*, where we cover the concept of sizing, the key information that you require to perform sizing exercises, and go through the sizing calculations based on the sizing tool.
- *Solution deployment and configuration*, where we draw from our experiences and discuss how to deploy and configure a CommonStore solution.
- *Journaling*, where we introduce the journaling feature provided by Lotus Domino and Exchange Server. Journaling becomes mandatory when there is a requirement to archive all e-mails passing through your company's e-mail servers.
- *eMail Search installation and configuration*, where we discuss what is eMail Search, the business requirement, and how to install and configure eMail Search for CommonStore.

- *Maintenance*, where we describe the maintenance tasks necessary to keep a CommonStore solution up and running smoothly and efficiently. To ensure a successful solution setup and continued performance after the solution is deployed, there should be an effective maintenance, backup, and recovery plan for the CommonStore solution. We outline some of the tasks that are necessary to maintain an efficient solution.
- *Logs, traces, and report logging feature*, where we list the important logs and traces that are generated by CommonStore, and introduce the new report logging feature.

It is not easy to collect and describe all the best practices for setting up CommonStore solutions. We faced many challenges in obtaining information from many sources. During the writing of this IBM Redbook, we interviewed the specialists and experts who currently practice in these areas, learnt from their experiences, and collected the best practices, hints, and tips from them. We also relied on the help of the developers and support from the IBM Germany team, and the materials written by them.

This IBM Redbook provides the starting point for a successful CommonStore solution setup. We share some of the known practices and knowledge with the readers. By including topics such as solution planning, sizing and design, deployment and configuration, journaling, maintenance, troubleshooting, and logs and traces in this IBM Redbook, we want you to be aware that these are the areas that impact your solution design and set up; these are the areas where you can make a difference. We encourage you to further explore, in a more in-depth manner, these areas as you start planning and implementing CommonStore solutions. If you find additional techniques and information along the way, we invite you to share your knowledge with us using IBM Redbook e-mail address or IBM intranet wiki:

https://w3.webahead.ibm.com/w3ki/display/CSBP

In future, when we have the opportunity to update this IBM Redbook, we can consolidate your suggestions in the IBM Redbook and provide more value to readers such as you.

The team that wrote this IBM Redbook

This IBM Redbook was produced by a team of specialists from around the world working at the IBM Development Lab in Germany.

Wei-Dong Zhu is a Content Management Project Leader with the International Technical Support Organization (ITSO) in San Jose, California. She has more than 10 years of software development experience in accounting, image workflow processing, and digital media distribution (DMD). Her development work with one of the DMD solutions contributed to a first time ever win for IBM of a technical Emmy award in 2005. She holds a Master of Science degree in Computer Science from the University of the Southern California. Jackie joined IBM in 1996. She is a Certified Solution Designer for IBM Content Manager.

Satya Bhamidipati is a Certified IT Software Specialist from the IBM Software Group with focus on Content Management suite of products. He has over 10 years of experience in the IT industry. He holds a Master's degree in Information Systems from the Arizona State University.

Elena Bogdanova is an IT Specialist with the IBM Information Management team in Moscow, Russia. Elena graduated with honors from Moscow Technical University with a Bachelor's degree in Mathematics Modeling. She has three years of experience in content management field. Her areas of expertise include Content Manager, CommonStore, Document Manager, and integration with IBM WebSphere® Portal and IBM Workplace™ Forms.

Toralf Gerstaecker is an IBM Software Group IT Specialist in Germany. He has 10 years of experience in Lotus Notes®/Domino. He has worked at IBM for seven years. His areas of expertise include large Domino enterprise infrastructures, enterprise content management solutions, and Domino monitoring.

Ryan Lott is the Technical Director of Neocol UK Ltd, an IBM Business Partner in Europe specializing in data management solutions. Ryan is an IBM Certified Solution Designer in Content Manager with five years of experience in technical presales, sizing, supporting, and deploying solutions based on Content Manager, Document Manager, and CommonStore.

Donald L Wilcox Sr. is a Certified Senior Software IT Specialist from the IBM Software Group with focus on the Content Management suite of products. He has more than 24 years of experience with IBM, with the last 10 years working in the Content Management area. Donald holds a Master's degree in Computer Information System from the University of Phoenix. He has experience with IBM MVS™, IBM AIX®, IBM DB2®, and WebSphere. His areas of expertise include design, implementation, and tuning Content Management applications.

Special thanks to **Ralf Hauser** for his contribution of the materials on the report logging feature.

Special thanks to **Monika Matschke** for her contribution of the materials on logs and traces.

Special thanks to **Bernhard Hailer** for his leadership and support throughout this IBM Redbook project.

Thanks to the following people for their contributions to this IBM Redbook:

Matthias Bierbrauer
Barbara Hartel
Jürgen Maletz
Stephan Peters
Eberhard Schoeck
IBM Software Group, CM & Discovery Development, IBM Germany

Guo Yan
Shen Hai Wei
Wen Nan Nan
Zhang Hui
Zhu Bing
IBM Software Group, CM & Discovery Development, IBM China

Reinhold Engelbrecht
IBM Sales & Distribution, CM & Discovery, IBM Australia

Holger König
IBM Software Group, Content Management Center of Excellence, IBM Germany

David Lebutsch
IBM Software Group, CM & Discovery Lab Services, IBM Germany

Martin Lorenz
IBM Sales & Distribution, IT Specialist Techline, IBM Germany

Peter Eichhorn
IBM Software Group, CM & Discovery, IBM Switzerland

James Merry
IBM Software Group, CM & Discovery Lab Services, IBM U.S.

Ken Nelson
IBM Software Group, CM & Discovery Customer Success Team, IBM U.S.

Robert Nonnenkamp
IBM Software Group, TechWorks, IBM U.S.

Thanks to the following people who have provided much needed graphic support and administrative support for this IBM Redbook project:

Emma Jacobs
Michael B Schwartz
ITSO, IBM U.S.

Helena Jeremen
IBM Software Group, Boeblingen, IBM Germany

Become a published author

Join us for a two- to six-week residency program! Help write an IBM Redbook dealing with specific products or solutions, while getting hands-on experience with leading-edge technologies. You'll have the opportunity to team with IBM technical professionals, Business Partners, and Clients.

Your efforts will help increase product acceptance and customer satisfaction. As a bonus, you'll develop a network of contacts in IBM development labs, and increase your productivity and marketability.

Find out more about the residency program, browse the residency index, and apply online at:

ibm.com/redbooks/residencies.html

Comments welcome

Your comments are important to us!

We want our IBM Redbooks™ to be as helpful as possible. Send us your comments about this or other IBM Redbooks in one of the following ways:

- Use the online **Contact us** review IBM Redbook form found at:

  ibm.com/redbooks

- Send your comments in an email to:

  redbooks@us.ibm.com

- Mail your comments to:

  IBM Corporation, International Technical Support Organization
  Dept. HYTD Mail Station P099
  2455 South Road
  Poughkeepsie, NY 12601-5400

1

Overview of CommonStore

This chapter provides an overview of the products involved in the CommonStore solutions.

We cover the following topics in this chapter:

- IBM CommonStore for Lotus Domino
- IBM CommonStore for Exchange Server
- Overview of IBM Content Manager
- Overview of eMail Search

1.1 Introduction

In today's business environment, e-mail has become one of the most commonly used forms of corporate communication. It is one of the core tools to conduct business transactions, and it generates a significant volume of critical business information. The types of information in e-mails include intellectual property, personal information, corporate announcements or broad communications, and business decisions.

Information Technology (IT) departments are under increasing pressure to establish secure archive management systems that can protect these business-critical assets and deliver information as needed to users, meet the demands of regulatory compliance and legal discovery, and enhance the speed and responsiveness of productions systems.

IBM CommonStore for Lotus Domino (CSLD) and IBM CommonStore for Exchange Server (CSX) are the message archival and retention management products for Lotus Domino and Microsoft® Exchange to manage the growth of e-mail systems. These mail management systems can be records-enabled using IBM Records Manager to retain and dispose e-mail records based on regulatory and legal requirements. IBM eMail Search (eMS) can also be used to help create a solution that is designed for legal discovery.

The focus of this IBM Redbook is on the e-mail message archival and retention solutions using CSLD, CSX, and IBM eMail Search. We address the following business issues:

- *Mailbox management issues*: Organizations are dealing with the growing volumes of e-mails due to the growing number of users, attachments, and e-mail size. Generally, e-mails are kept for a long period of time. This increases the backup and recovery times for mail servers, the storage requirements, and the processing time required to build indexes.

  CommonStore solutions help to:

  – Keep mail database size within defined limits
  – Reduce backup and recovery times
  – Reduce storage costs

- *Long-term retention and discovery issues*: Organizations spend a lot of time searching for e-mails across mailboxes when involved in a legal dispute. Therefore, it is necessary to retain the e-mails for long term and provide search capability on the e-mail content as required.

  CommonStore solutions with eMail Search provide the following advantages:

  – Help organization to meet legal obligations
  – Remove old or obsolete e-mails

The CommonStore solutions that we address in this IBM Redbook consists of the following products and components:

- IBM CommonStore for Lotus Domino for Domino mail systems
- IBM CommonStore for Exchange Server for Exchange mail systems
- eMail Search for long-term e-mail retention and discovery purpose

We examine these products and components in the following sections.

1.2 IBM CommonStore for Lotus Domino

IBM CommonStore for Lotus Domino (CSLD) is an archiving application for IBM Lotus Notes documents. It can archive Lotus Notes document content including attachments from any Lotus Notes database.

CommonStore for Lotus Domino provides the following functions:

- Migrating document content or folders from Notes databases to an archive
- Searching for content in an archive
- Displaying archived content
- Retrieving or restoring archived content
- Deleting archived content

1.2.1 E-mail and archive repository definition

An e-mail message consists of:

- Message body
- Message properties (visible or invisible)
- Attachments (optional)

An *archive* is a repository that serves as a container for archived e-mails. CommonStore for Lotus Domino works with any one of the archive repositories provided by the following products:

- IBM Content Manager
- IBM Content Manager OnDemand
- IBM Tivoli® Storage Manager

1.2.2 E-mail archiving with CommonStore for Lotus Domino

CommonStore for Lotus Domino offers e-mail archive, search, retrieval, viewing, and deletion functions. Some of these functions have to be run manually and others can be done automatically. Automatic functions are set up centrally and include policy-driven archiving and administrator-triggered retrieval.

Administrator-triggered retrieval is for retrieving a large number of documents. Manual functions are added to the users' mail databases by modifying the mail database templates. Some manual functions are possible without mail template modifications, for example, the optional function to retrieve via an embedded hotspot. With Lotus Notes documents, users can archive information about the document body, attachments, or information in other fields.

E-mail archiving types

CommonStore for Lotus Domino offers a variety of e-mail archiving types:

- *Attachments*: Archives only the e-mail attachments.
- *Entire*: Archives and saves the entire e-mail as Notes native format or DXL format.
- *Component*: Decomposes the e-mails into separate components and archives the entire e-mail as components.

You can also convert and save content as American Standard Code for Information Interchange (ASCII), Rich Text Format (RTF), or other formats.

E-mail deletion after archiving

After an e-mail is archived, you can set up various ways for CommonStore to delete e-mails from the original mail databases. The deletion type includes:

- *Nothing*: Does nothing; leaves the original e-mail in the mail databases.
- *Attachments*: Removes the e-mail attachments and replaces them with hyperlinks.
- *Body*: Removes the e-mail body content and the attachments; replaces them with message stubs and links to the archived content. This option is only available with Entire and Component types of archiving.
- *Message*: Removes the entire e-mail message; no stub is provided. This prevents users from accessing the archived e-mail directly.

For example, to reduce the size of mailboxes, a Notes administrator sets up an archiving policy: Archiving and then delete all attachments after 30 days (archiving type: Attachment) and archive and delete the entire mails after 90 days (deletion type: Message). In this example, a user sees the entire e-mail along with the attachments for 30 days. After 30 days, the archival policy archives and deletes the attachments and places a hyperlink in place of the attachments. The user can still see the body of the e-mail, but has to use the hyperlink to view the attachments. After 90 days, the e-mail content (including its attachments) is archived and removed from the users' mailbox. If the user wants to access the e-mail, a search is run and the e-mail is retrieved from the search results.

E-mail archiving and deletion can be done manually or automatically.

1.2.3 CommonStore for Lotus Domino system architecture

CommonStore for Lotus Domino consists of the following components:

- Archpro
- Archive agents
- Domino dispatcher
- CSLD task
- CSLD crawler (Notes agents)
- CSLD configuration database (Notes database)
- CSLD job database (Notes database)
- HTTP dispatcher

Figure 1-1 shows the CommonStore for Lotus Domino system architecture.

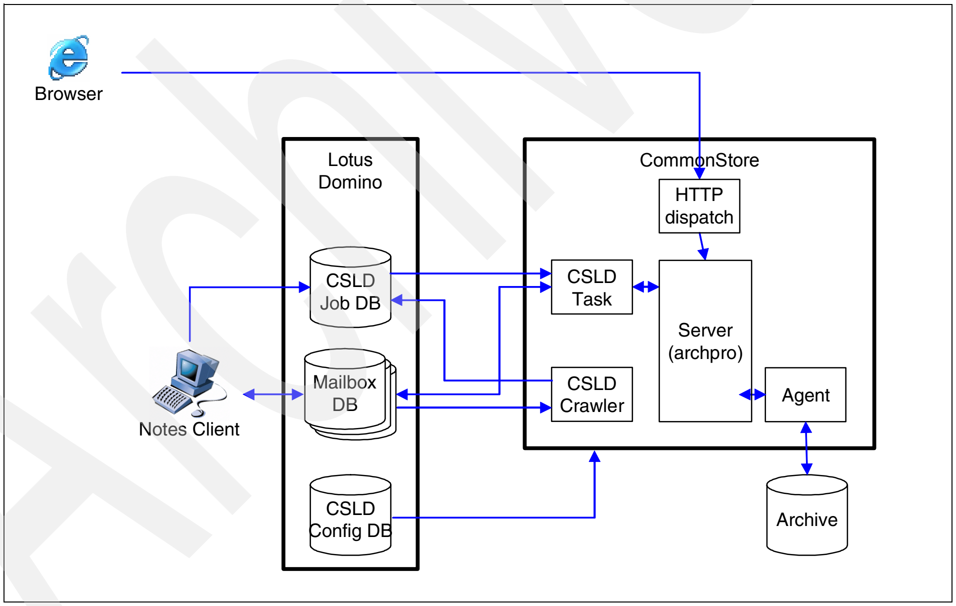


Figure 1-1 CommonStore for Lotus Domino system architecture

Archpro

The archpro program is the central part of CommonStore for Lotus Domino. It maintains a list of archives it is connected to, and controls the flow of information to and from the archives. All input and output is routed through the archpro program. Whenever e-mail content is to be archived, an archiving request is sent to the archpro program, along with the content and the content's descriptive information.

The archpro program does not handle direct communication with the archive itself. It maintains a queue for archive-dependent agents, such as the Content Manager agents, and load balances all requests among these agents.

Archive agents

An archive agent is the interface to the archive. Every agent is an independent archive client process. For every archive supported by CommonStore for Lotus Domino, there is a special agent. An agent calls the application programming interfaces (APIs) of the archive. For example, when the back-end archive uses Content Manager, the agent calls the Content Manager client APIs.

Multiple agents for the same archive can run in parallel. An agent gets its orders from the archpro program. When all agents are busy, the request is held in the queue of the archpro program until an agent becomes available. Every agent keeps one connection to the archive. The more the number of agents, the more requests can be processed concurrently. Every agent requires system resources, therefore the number of concurrent agents is limited by the server's resources. Agents always run on the same machine as the archpro program and are automatically started by the archpro program.

Domino dispatcher

The Domino dispatcher (not shown in the diagram) is the interface between the archpro program and the CommonStore for Lotus Domino task (which we explain later in this chapter). It translates Domino requests into the language of the archpro program. The Domino dispatcher is a multi-threaded application allowing multiple CommonStore for Lotus Domino tasks to connect to it and send requests. It runs on the same machine as the archpro program, and is automatically started by the archpro program.

CSLD task

The CSLD task is a server process that periodically polls jobs from the CSLD job database (which we explain later in this chapter). *Jobs* are requests to CommonStore for Lotus Domino. The CSLD task converts Notes documents to files that are archived and vice versa. According to the request stated in the job, it sends the request along with the file to the Domino dispatcher, which in turn forwards it to the archpro program. In CommonStore for Lotus Domino, there are archiving tasks and retrieval tasks. Archiving tasks process requests that add or modify content in the archive (for example, archiving, updating, or deleting e-mails). There are retrieval tasks that process requests that retrieve e-mail content from the archive (for example, searches or single document retrieval). CSLD tasks are not Domino server add-on tasks, but stand-alone applications. Multiple tasks can run on one machine.

CSLD crawler

The CSLD crawler is the program (Notes agent) that carries out the automated functions of CommonStore for Lotus Domino. The crawler is a Notes agent that gets its configuration from the CSLD configuration database (which we explain later in this chapter) on a Domino server. It goes through the Notes databases matching the configured criteria, and creates archive or deletion jobs in the CSLD jobs database (which we explain later in the chapter) on the Domino server. This independent program is the central application for the following functions:

- Policy-driven archiving
- Policy-driven deletion
- Administrator-triggered retrieval

When used for policy-driven archiving or administrator-triggered retrieval, the crawler checks whether documents in the specified Lotus Notes databases meet a set of selection criteria, which is called a policy. If documents meet these criteria, the crawler creates archiving or deletion jobs for them.

CSLD configuration database

The CSLD configuration database is a Lotus Notes database and is located on the Domino Server. It contains the configuration documents and the policies required for e-mail archiving. The schedules, document selection criteria, and archiving polices are defined in the configuration database.

CSLD job database

The CSLD job database is a Lotus Notes database and is located on the Domino server. It collects all requests, such as archiving, retrieval, search, and viewing, and passes them onto the CSLD task.

HTTP dispatcher

The HTTP (Web) dispatcher is used for viewing archived content in a Web browser. The Web dispatcher runs on the same machine as the archpro program, and is automatically started by the archpro program.

1.3 IBM CommonStore for Exchange Server

IBM CommonStore for Exchange Server (CSX) is an archiving application for Microsoft Exchange documents. It can archive Exchange e-mail messages including attachments from any Exchange mailbox that resides on an Exchange Server.

CommonStore for Exchange Server provides the following functions:

- Migrating e-mails from Exchange mailboxes or public folders to an archive
- Searching for content in an archive
- Displaying archived content
- Retrieving archived content
- Deleting archived content

1.3.1 E-mail and archive repository definition

An e-mail message consists of:

- Message body
- Message properties (visible or invisible)
- Attachments (optional)

An *archive* is a repository that serves as a container for archived e-mails. CommonStore for Exchange Server works with any one of the archive repositories provided by the following products:

- IBM Content Manager
- IBM Content Manager OnDemand
- IBM Tivoli Storage Manager

1.3.2 E-mail archiving with CommonStore for Exchange Server

Similar to CommonStore for Lotus Domino, CommonStore for Exchange Server offers e-mail archive, search, retrieval, viewing, and deletion functions. Some of these functions have to be run manually and others can be done automatically. Automatic functions are set up centrally and include policy-driven archiving.

E-mail archiving types

CommonStore for Exchange Server offers a variety of e-mail archiving types:

- *Attachments*: Archives only the e-mail attachments, this includes embedded attachments.
- *Entire message*: Archives the entire e-mail including message body, properties, and attachments.

- *Envelope entire*: Supports Microsoft envelope journaling and is a specialized form of entire message archiving.
- *Component*: Decomposes the e-mails into separate files, one file for each component and one MSG file for the message body.

E-mail deletion after archiving

After an e-mail is archived, you can set up various ways for CommonStore to delete e-mails from the original mail databases. The deletion type includes:

- *Nothing*: Does nothing; leaves the original e-mail in the mail databases. Nothing (no link) is a specialized type of deletion and is designed for the purpose of archiving envelope of journal messages.
- *Attachments*: Removes the e-mail attachments and replaces them with hyperlinks.
- *Body*: Removes the e-mail body content and the attachments; replaces them with message stubs and links to the archived content. This option is only available with archiving type of Entire and Component.
- *Message*: Removes the entire e-mail message; no stub is provided. This prevents users from accessing the archived e-mail directly.

E-mail archiving and deletion can be done manually or automatically.

1.3.3 CommonStore for Exchange Server system architecture

CommonStore for Exchange Server consists of the following components:

- Archpro
- Archive agents
- CSX System Manager
- CSX task (Crawler, Poller, Worker, and Committer)
- Search server
- CSX configuration folder
- CSX job folder
- Web (HTTP) dispatcher

Figure 1-2 shows the CommonStore for Exchange Server system architecture.

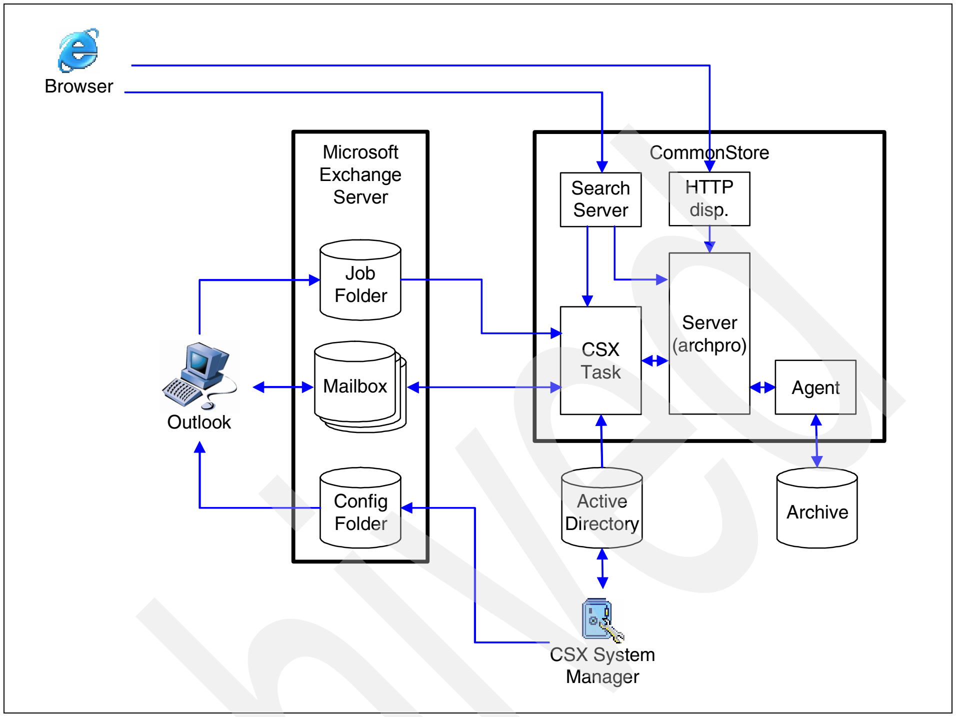


Figure 1-2 CommonStore for Exchange Server system architecture

Archpro

The archpro program is the central part of CommonStore for Exchange Server. It maintains a list of logical archives and controls the flow of information to and from these archives. All input and output data is routed through the archpro program. Archpro distributes the requests among the archive agents, which are the interfaces to the archive.

Archive agents

An archive agent is the interface to the archive. Every agent is an independent archive client process. For every archive supported by CommonStore, there is a special agent. An agent calls the APIs of the archive. For example, when the back-end archive uses Content Manager, the agent calls the Content Manager client APIs.

Multiple agents for the same archive can run in parallel. An agent gets its orders from the archpro program. When all agents are busy, the request is held in the queue of the archpro program until an agent becomes available. Every agent keeps one connection to the archive. The more the number of agents, the more requests can be processed concurrently. Every agent requires system resources, therefore the number of concurrent agents is limited by the server's resources. Agents always run on the same machine as the archpro program and are automatically started by the archpro program.

CSX System Manager

The CSX System Manager is a Microsoft Management Console (MMC) that provides a graphical user interface for the administration of CommonStore for Exchange Server. Using the CSX System Manager, you can configure the working of your archiving solution including the instances of the CSX task. The CSX System Manager saves the settings in the Active Directory® of the forest to which the CSX task instances and the connected Exchange servers belong to. When an instance of the CSX task is started, it reads the configuration data from this Active Directory. Additionally, the CSX System Manager writes client-relevant information into the configuration folder.

CSX task

The CSX task is the program that directly interacts with your Microsoft Exchange environment. It is responsible for performing interactive and automatic archiving and retrieval, during which it transforms Exchange messages into files and vice versa.

The CSX task can be logically split into the following components:

- Crawler
- Poller
- Worker
- Committer

Figure 1-3 shows the individual components of the CSX Task.

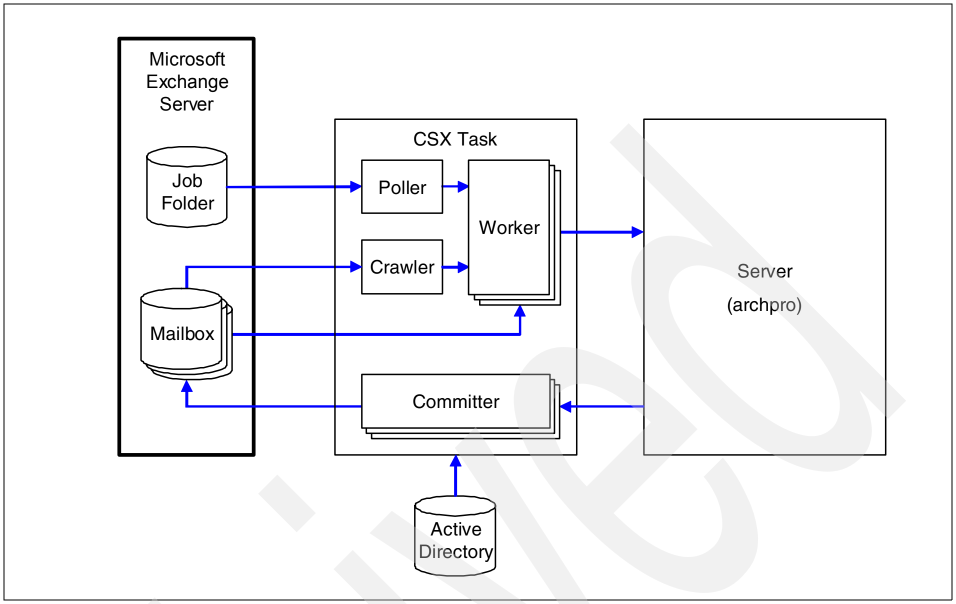


Figure 1-3 CSX task components

CSX Task: Crawler

The crawler component performs policy-driven archiving processes. It investigates mailboxes and public folders to see whether any message meets certain selection criteria. These criteria are defined in policies. If the messages meet the criteria that you defined in policies, the crawler creates archiving requests for them. Each crawler does this for exactly one Exchange Server.

In addition, the crawler initiates the automatic removal of retrieved content, also known as restubbing. More discussion about restubbing is provided in “CSX Task: Committer” on page 13.

CSX Task: Poller

The poller component continually looks for interactive job messages in the job folder. Interactive jobs are created by clients (Outlook® users). The job messages contain the information necessary to perform the jobs. The poller converts this information and puts it in an internal job queue.

CSX Task: Worker

The worker component reads the instructions from the converted job messages in the internal job queues. It then accesses the Outlook messages referred to in the job messages and reads the content to be archived or identifies the content to be retrieved. It finally creates the job request and passes it to archpro.

CSX Task: Committer

The committer component receives the answers from archpro and processes the instructions within it. If an operation is successful, this results in modifications of the original messages according to the instructions. For archiving jobs, this means that content might be deleted from the original messages according to the selected deletion type. Additionally, the status is set to archived and a few properties are added to the message. For retrieval jobs, it means restoring archived content and setting the status to retrieved. The committer is also responsible for restubbing.

The term restubbing describes the process of deleting retrieved content after a certain time and resetting the status to archived. When messages are restubbed, this is done in accordance with the deletion type that is specified at the time that the message is archived. Restubbing does not involve communication with archpro.

CSX configuration folder

The configuration folder contains all of the information that Microsoft Outlook clients require to run the CommonStore functionality. This includes:

- Job folder name
- Server host names and ports

This data is written during system configuration using the CSX System Manager.

CSX job folder

The job folder contains interactive client requests to archive or retrieve messages. They are collected in the folder before they are picked up by a CSX task.

Search server

The search server is a separate Web application server process for the search function. It is automatically started by the archpro. It routes the search requests through the archpro to the archives that support this function (Tivoli Storage Manager does not support it). After receiving the results from the archpro, it arranges them into a list and passes that list to the requesting Web application that shows it in the browser window. The result list enables the requesting user to view the messages and attachments that are found in a browser. In addition, the user can restore the archived content to a special folder in the user’s mailbox.

Messages are restored to their original form, and attachments are restored to container messages.

Web (HTTP) dispatcher

The Web (HTTP) dispatcher is a Web server for content archived by CommonStore. You can use it to view the archived content in a Web browser. The Web dispatcher is installed as part of the archpro and is automatically started by it.

1.4 Overview of IBM Content Manager

IBM Content Manager is one of the back-end archive repositories supported by CommonStore for Lotus Domino and CommonStore for Exchange Server. In this IBM Redbook, we focus on Content Manager as the repository for CommonStore. Content Manager is built on a multi-tier distributed architecture, with a Library Server that manages, indexes, and searches documents, and Resource Managers that manage the objects.

Client applications use APIs to start all Content Manager services, which are divided between the Library Server and one or more Resource Managers. A single Content Manager implementation supports a single Library Server, along with one or many Resource Managers and clients.

1.4.1 Library Server

The Library Server manages the content metadata and is responsible for access control to all content. It maintains the indexing information for all multimedia content held in a Resource Manager. Users submit requests through the Library Server. The Library Server validates the access rights of the requesting client and authorizes the client to directly access the object in the designated Resource Manager. The Library Server also maintains referential integrity between the indexing information and the objects themselves. It is built on relational database management system. All access to the Library Server is through the database query language Structured Query Language (SQL), and all Library Server logic runs within database. With Content Manager, no persistent processes operate on the Library Server; all content management functions are stored procedures run by the database. Content metadata in the Library Server is backed up and recovered using standard database tools.

1.4.2 Resource Manager

Resource Managers are the repositories that contain the digitized content and manage the storage and retrieval of objects. The Resource Manager supports caching and replication, and provides hierarchical storage management when used in conjunction with IBM Tivoli Storage Manager. The Resource Manager architecture provides an extensible model that enables the support of additional Resource Managers in the future.

1.5 Overview of eMail Search

IBM eMail Search (eMS) for CommonStore provides the discovery feature that helps organizations to respond to litigation or regulatory agency requests.

eMail Search client adds an easy-to-use Web interface that specific users (compliance and litigation officers) can use to perform general discovery steps on e-mails stored in the Content Manager by CommonStore.

eMail Search provides the following functions:

- Helps authorized personnel (for example, legal department and compliance officers) to search across archived mailboxes.
- Locates business-critical e-mails through provided search features; for example, you can search based on the e-mail fields, such as to, from, cc, and attachment content.
- Helps to respond to requests to demonstrate adherence to corporate-compliance plans.
- Helps to respond to legal requests for e-mail record discovery requests associated with litigation.
- Creates result sets of e-mail records for analysis or transfer.

eMail Search for CommonStore for Lotus Domino and Exchange Server provides a discovery feature that you can use to find e-mail records in response to litigation-related requests. It offers an e-mail archiving solution to manage the entire life of an e-mail, from creation to disposition, and seamlessly integrate every stage of the process, including:

- Flexible capture
- Scalable archive repository
- Messaging system volume management
- Controlled retention and destruction
- Wide spectrum of secure, long-term storage device support
- Access to archive using e-mail search and discovery

We discuss eMail Search in more detail in Chapter 7, “eMail Search for CommonStore” on page 107.

2

Solution planning

This chapter discusses the solution planning for an IBM CommonStore solution based on client requirements. The insight can help clients, project managers, sales people, and even technical staff to avoid common pitfalls during the planning and setup phase of an implementation project of a CommonStore solution.

We cover the following topics in this chapter:

- Requirement gathering and management
- Sample project plan
- Main deliverables of the solution planning

The main focus of this chapter is a CommonStore solution independent from the e-mail system in use. Therefore, we distinguish between CommonStore for Lotus Domino (CSLD) and CommonStore for Exchange Server (CSX) only if necessary.

This chapter is not intended to provide an overview of certain subjects of project management nor does it promote a single method. It shows some examples of what we encountered during similar projects.

2.1 Requirement gathering and management

We describe common client requirements and the process of gathering and managing these requirements. We also provide an overview of the implications that the requirements have on project planning and implementation.

Before deep diving into technical features of software products that make up a solution such as CSLD and Content Manager, a sufficient process of requirements gathering and management must be established. A well-known risk is to start with implementing parts of the solution before deciding what the requirements are and how these requirements are reflected in the solution. However, it is almost never possible to define a full set of requirements at the beginning of the project. Therefore, we recommend that you maintain a list of assumptions, which contains all issues that cannot be decided at the beginning of solution planning such as archiving format and retention periods, or issues that cannot be defined in detail as required such as network bandwidth or data volumes to archive. The goal of requirement management is also to minimize the entries in the list of assumptions.

2.1.1 Requirement workshop

A common approach is to conduct a requirements workshop to plan a CommonStore solution. This is the place where requirements mainly from a business point of view are gathered. However, it is advisable to invest a reasonable amount of time in judging the technical feasibility of the CommonStore solution based on these business requirements. There are three main areas of requirements that you should consider:

- Mailbox management
- Long-term retention and discovery (compliance)
- Message monitoring and supervision

In the remainder of this IBM Redbook, we discuss the first two areas, because these are the most common ones.

Introducing requirements for a CommonStore solution

In the last few years, we have seen a change of drivers for implementing CommonStore solutions. Five years ago, the main driver for CommonStore solutions was mailbox management with a focus on reducing disk space that is consumed by mail files, addressing the growing mail volume, and moving older e-mails to cheaper storage. This has changed dramatically. In the last two years, almost every business is facing the demands of compliance regulations and the ongoing integration of e-mail into critical business processes. It is common sense these days that an e-mail system has its costs and that saving or avoiding costs is not a business driver. Business drivers are the adherence to regulatory and company rules, meeting compliance criteria set by internal or external instances, and the usage of e-mail to create business value.

The biggest challenge to achieving e-mail compliance is to understand the regulations that are applicable and their impact on a CommonStore solution. The current political focus on regulatory compliance and the raft of new regulations coming into effect in almost every country makes it challenging to ensure that existing compliance policies are up-to-date. This makes the discussion of compliance a large part of the requirement workshop.

E-mail communications are generally affected by a number of different internal and external regulations and policies. It is important when defining e-mail regulatory compliance policies to ensure that maximum synergy is achieved for the outlined requirements. In some cases, such as where numerous industry and geography specific regulations are applicable, it might be wise to seek specialist advice. Before starting with the requirements workshop, it might be even advisable to conduct an internal audit of the existing compliance environment and to contrast with the required compliance environment. The resulting gap analysis reveals compliance exposures and helps to define the new or adapted polices that are required.

Another fact that makes matters complex is that e-mail is personal, asynchronous, and unstructured. Some people even use their mail file as their primary document repository. Mail files are always a source of a huge amount of intellectual capital and process or organizational knowledge. However, e-mail was never designed for such uses. You should have all these circumstances in mind when you plan a CommonStore solution.

It is common for different groups within the same organization to have differing requirements. These can greatly increase the complexity and cost of delivering the required CommonStore solutions. It is advisable to significantly reduce the number of separate policies by using a modular approach with a limited number of variations. We recommend a top-down approach to firstly fulfill the common, major requirements and secondly to review and to fulfill the special, minor requirements. Information Technology (IT) has a key role to play in assisting clients to assess the impact of compliance within their organization by providing tools to support and monitor the required CommonStore solutions. These tools help to reduce the costs associated with compliance by automating manual processes and reducing delivery time by improving collaboration between the many groups (internal and external) that are vital for success. Compliance tools can also help reduce risk by providing management with an improved visibility of the compliance process and by helping to avoid human error.

Workshop contents and participants

In most of the situations, a single requirement workshop is not satisfactory. Two or more workshops might be necessary. The first (series of) workshop should be about business requirements including compliance policies. Involve the following parties:

- Legal departments or departments dealing with compliance rules for the company
- Security and contingency
- IT architects
- Service provider representatives (knowledgeable of the mail system)

After or during the definition of business requirements, another important subject of the planning phase of a CommonStore solution is an assessment of the current IT environment, which is the basis for further technical planning. The assessment document must contain information about hardware, software versions, network bandwidth, locations, and data volumes. There are some questionnaires that are available from IBM, which we discuss in 3.3, “Requirement-gathering questionnaires” on page 30.

For further management of requirements and to provide a means of cross-checking the CommonStore solution at any time, a set of use cases can be helpful. At least the main processes should be available as use cases with which it is always possible to rate the quality of the solution and the project progress.

Functional and nonfunctional requirements

The main deliverable of the requirement workshops is a document containing the functional and nonfunctional requirements for the CommonStore solution.

Functional requirements define the internal workings of the CommonStore solution, that is, the calculations, technical details, data manipulation and processing, and other specific functionality that show how the use cases are to be satisfied. They are supported by nonfunctional requirements, which impose constraints on the design or implementation (such as performance requirements, quality standards, or design constraints).

The core of the requirement is the description of the required behavior, which must be clear and readable. This behavior might come from organizational or business rules, or it might be discovered through elicitation sessions with users, stakeholders, and other experts within the organization. Many requirements can be uncovered during the use case development. When this happens, the person responsible for the requirement analysis should create a placeholder requirement with a name and summary, and research the details later, to be filled in when they are better known.

Some examples for the functional requirements for a CommonStore solution are (the examples are common and require further specification in a real planning situation):

- Archiving of e-mails in a required, unchangeable format.
- Central archiving and deletion policy management (might include records management).
- Make critical business documents centrally stored and available even after years.
- Provide the means for search and retrieval of e-mails.
- Provide management reporting on adherence of compliance rules and other regulations.
- Integrated solution for current business processes.
- Usage of different clients, also independent clients that are not part of the mail system.
- Usage of dedicated clients for search and retrieval.
- Search on e-mails using full-text, different categories of documents for individual users, or both.
- Automatically offload e-mails from user mail files.
- Support for mobile and offline users.
- Manual options for categorizing and archiving.
- Reduce the size of user mail files.

In systems engineering and requirements engineering, nonfunctional requirements are requirements that specify criteria that you can use to judge the operation of a system, rather than specific behaviors. Contrast this with functional requirements that specify specific behavior or functions. Typical nonfunctional requirements are reliability, scalability, and cost. Nonfunctional requirements are often called the *ilities* of a system. Other terms for nonfunctional requirements are *quality attributes* and *quality of service requirements*.

Some examples for nonfunctional requirements for a CommonStore solution are:

- Availability (certain service level agreements)
- Usability by target user community
- Documentation
- Legal and licensing issues
- Maintainability

- Platform compatibility
- Quality (for example, faults discovered, faults delivered, fault removal efficacy)
- Resource constraints (for example, processor speed, memory, disk space, network bandwidth)
- Response time and performance (for example, run time, response time, volumes of records and transactions)
- Security

2.1.2 Requirement management

Typically, you can maintain a list of requirements containing their status. We recommend that you review this list frequently and discuss new or changed requirements. This list is also influenced by the technical feasibility of a CommonStore solution. Depending on the established standard for requirements management or for project management in general, different methods apply to this task. Some examples are:

- IBM Global Services Method
- Institute of Electrical and Electronics Engineers (IEEE) standards for software specification
- Accelerated Value Method based on value frames

It is not important which method you use for the CommonStore solution, but it is essential to follow the rules of the selected method and to use the tools that the method provides.

For instance, the worst case for not properly managing requirements can be a ready deployed pilot system, which is already filled with production data, and in this phase someone from the business team wants an additional field to be added in the Content Manager index.

Often the requirements for a CommonStore solution have an impact on user experience and functionality of the e-mail system. Users must be aware that archived documents have to be searched, retrieved, or both before the data can be used further. Beyond that it is a demand to deal with offline users and locally saved data, such as notebook users with local replicas of their mail files.

Another example of the impact of a CommonStore solution is the full-text-search feature of the e-mail system, especially when it is Lotus Notes. Within Lotus Notes, users can create a full-text index of all the information contained in their mail files. After archiving documents from the mail file, there is some information for building the full-text.index that is missing. This leads to a certain amount of feature loss because a full.text.search query does not deliver the same results as it does before archiving.

The main objective of the requirements management task is to handle such changes in user experience and functionality.

2.1.3 Requirements versus expectations

There is a difference between requirements and expectations. The main expectation is often that a CommonStore solution can be implemented out-of-the-box and does not require much planning. However, we learned in many projects that there is always a certain degree of customizing that is involved. A good example is the Domino mail template, which is customized in almost every project.

Some of the lessons learned from our projects:

- Do not underestimate the time required for testing.
- E-mail archiving within the context of information lifecycle management (ILM) is a very complex task.
- A CommonStore solution is not likely to be operated by the already existing staff members in place. It requires adjustments in terms of the number and organization of administrative staff. Additionally, the establishment of new interfaces might be necessary, for example, to the legal department.

2.2 Sample project plan

In this section, we discuss a sample project plan containing the main tasks of an implementation project. It shows the relations between the several project tasks and highlights best practices from the field.

A typical structure for a project plan can be:

- Assessment phase
  - Current IT environment assessment
  - Requirements workshop
  - Functional and nonfunctional requirements
  - Use cases
- Design phase
  - High-level solution design
  - Sizing

- Planning phase
  - Hardware and location readiness
  - Test plan
  - Migration plan
  - Deployment plan
- Build phase
  - Customizing
  - Hardware build
  - Software installation
- Test phase
  - Conducting tests
  - Reviewing test results
- Deployment phase
  - Deploying the solution

We cannot emphasize enough how important the requirement gathering and management is. In the structure given previously, the requirement gathering and documenting is the main part of the assessment phase. (Refer to 3.3, "Requirement-gathering questionnaires" on page 30 for more information about requirement gathering.) Throughout the whole duration of the project, we recommend that you use prototyping and involve the business people early and extensively.

Unlike other solutions, there are two or three projects phases that bind a reasonable amount of the available resources: assessment and design. This can be a significantly larger share than in other projects. This is because the requirements can be so various and requirement management is an ongoing process. Another aspect can be the customizing of the Domino mail template if a CommonStore solution with CSLD is planned. It can involve a lot of interaction between the project team and the approvers of the mail file design.

It is a good practice to add the task of knowledge transfer into the project plan. As stated previously, the CommonStore solution requires more new skills. These new skills must be built by the operating and support staff and other people responsible for the solution in the early stages of the projects. If this is not possible, it is advisable to have several hand-over sessions for transferring the knowledge to support and operation personnel.

We provide two sample project plans in the additional material for this IBM Redbook.

2.3 Main deliverables of the solution planning

This section contains a listing of the solution planning deliverables.

The solution planning deliverables are:

- Assessment of the current IT environment
- Documentation of functional and nonfunctional requirements
- Review of compliance policies
- Risk analysis: Identification and mitigation recommendations
- Enterprise deployment schedule and labor estimate
- Detailed project plan
- Pilot definition
- Document hardware specifications
- Mail template design changes and documentation

In conclusion, even if you have to provide all of the deliverables listed previously, solution planning is an important project phase. Decent planning can ensure the success of your CommonStore solution, and is the foundation for all other project phases such as solution design and deployment.

3

Solution design

This chapter discusses the CommonStore solution design. We address the key areas to consider when planning and designing a solution using IBM CommonStore for Lotus Domino (CSLD) or IBM CommonStore for Exchange Server (CSX).

We cover the following topics in this chapter:

- Key information for CommonStore solution design
- Understanding solution components
- Requirement-gathering questionnaires
- Sample CommonStore solutions

The primary focus of this chapter is to detail the questions that you have to ask to get detailed requirements so that you can design the right system architecture for the e-mail archiving solution. This chapter covers both CSLD and CSX. We highlight the differences where necessary.

3.1 Key information for CommonStore solution design

The key information for CommonStore solution design are:

1. Understand all the components that comprise a CommonStore solution. Know what each component does and how it interacts with the other components from a high component level and in a detailed subcomponent level.
2. Gather requirements using questionnaires to:
   - Understand the primary requirement for the e-mail archiving solution.
   - Understand the existing mail system infrastructure.
   - Gather specific business requirements.
3. Perform sizing based on the input from the questionnaires and the sizing tool.
4. Map requirements to system architecture design.

From the requirements that you gathered from the questionnaires and the output from the sizing tool, map this information to architect the CommonStore solution. The results from a CommonStore solution design should include defining the following system architecture elements and decisions:

- The software components required for the solution.
- The hardware requirement (processing speed, memory, disk storage) for the software components based on the sizing output.
- The component topology; this includes:
  - Whether multiple CommonStore servers are required and how to pair (or distribute) the servers among the existing mail servers.
  - Whether multiple Resource Managers are required and how to set them up (for example in different locations).
  - Setting up the CommonStore server and Content Manager server (both Library Server and Resource Manager) on the same machine or on separate machines (or nodes).

    This depends partially on sizing and partially on available resource, preference, and performance requirement.
- Whether to enable journaling feature, and if yes, how to set it up.
- Whether to enable full-text search feature.
- Archiving policy and deletion (or retention) policy.

3.2 Understanding solution components

A CommonStore e-mail archiving solution provides an enterprise e-mail archiving solution that tightly integrates into your existing mail system environment. To design an efficient working solution, it is important to understand the components that make up a CommonStore solution and how they work with each other.

CommonStore is not a stand-alone solution. To make a CommonStore for Lotus Domino or a CommonStore for Exchange Server solution work, the software must connect to and combine two systems:

- Lotus Domino or Microsoft Exchange servers
- Back-end archive servers (such as Content Manager)

As a result, implementing CommonStore for Lotus Domino or CommonStore for Exchange Server includes not only configuring CommonStore, but also the Domino or Exchange server and the archive. CommonStore is considered a system integration or application gateway product for the two environments.

CommonStore for Lotus Domino and CommonStore for Exchange Server consist of the following main components:

- Archpro
- Agents
- CSLD or CSX tasks
- Crawler agents

To get an overview of these components, see 1.2.3, “CommonStore for Lotus Domino system architecture” on page 5, and 1.3.3, “CommonStore for Exchange Server system architecture” on page 9.

Content Manager (the archive repository that we focus on in this IBM Redbook) consists of the following components:

- Library Server
- Resource Manager

For an overview of these components, see 1.4, “Overview of IBM Content Manager” on page 14.

For CommonStore eMail Search, we provide an overview of eMail Search in 1.5, “Overview of eMail Search” on page 15, and more detailed discussion in Chapter 7, “eMail Search for CommonStore” on page 107.

For a more detailed explanation of these products and solutions, see the following publications:

- *CommonStore for Lotus Domino V8.3.1 Administrators and Programmer's Guide*, SH12-6742
- *CommonStore for Exchange Server V8.3.1 Administration & User's Guide*, SH12-6741
- *Content Manager Enterprise Edition V8.3: Planning and Installing Your Content Management System*, GC27-1332
- *DB2 Net Search Extender Administration and User's Guide V9.1*, SH12-6842
- *IBM Tivoli Storage Manager Implementation Guide*, SG24-5416
- *IBM WebSphere Application Server, Version 6.1 and IBM WebSphere Application Server Network Deployment, Version 6.1, data sheet*, G224-9140
- IBM eMail Search for CommonStore documentation (comes with the software)

3.3 Requirement-gathering questionnaires

In Chapter 2, “Solution planning” on page 17, we introduce the concept and information regarding requirement gathering. In this section, we reiterate some of the concepts and go into more details about requirement gathering process.

A requirement is an opportunity to address a business problem. A good requirement is one that is complete, consistent, correct, modifiable, traceable, unambiguous, prioritized, understandable, and verifiable. Gathering and evaluating requirements is an important exercise that:

- Reduces the likelihood that an inappropriate solution is implemented
- Exposes other problems that might also have to be addressed
- Drives the creation of the project acceptance criteria
- Reduces the likelihood of scope creep

Tips: Perform a thorough requirement-gathering exercise before the implementation of a solution. Document the results and ultimately use them to determine the satisfactory completion of the project.

Interview the business leaders and technical staffs. Ask the right questions and obtain the right background information and requirements. Accurate requirements are crucial in the design of the solution.

We present a set of questionnaires for background information and requirement gathering. We organize the questionnaires into three main categories:

- Primary solution requirement and the design implications
- Current mail system infrastructure and mail usage
- Specific business requirements

The answers to some of these questions also serve as input to the sizing tool that helps you to estimate the hardware required for your solution. For more details, see Chapter 4, “Sizing” on page 45.

For a complete set of the standard questionnaires that are used by the CommonStore lab services professionals, see Appendix A, “E-mail archiving questionnaires for solution planning and sizing” on page 183.

3.3.1 Primary solution requirement and the design implications

At the solution design and planning stage, you must get a clear picture of why there is a need to implement the solution. In other words, you have to understand the primary reason for implementing the e-mail archiving (CommonStore) solution.

There are two main reasons to implement a CommonStore solution:

- Mailbox management
- Long-term retention and discovery (compliance)

Based on the primary reason or requirement of the e-mail archiving solution, you can derive the software requirements that you require for the implemented solution.

Implementing the solution for mailbox management

Every year, e-mail system grows exponentially. You require sufficient infrastructure to manage e-mails.

CommonStore mailbox management solution helps you to archive old or unwanted e-mails from the production e-mail servers to the less expensive and less used external storage. The solution helps you to reduce the backup times of the active, frequently used e-mail data, optimize the data storage on the production servers, and minimize the associated operational costs. Implementing the mailbox management solution improves e-mail system usage and performance.

Solution design: Defining the software components

For a solution that is designed for mailbox management, you have to integrate the following software components into your existing mail server environment:

- *CommonStore* to archive old e-mails to the back-end repository: Use CSLD or CSX based on the existing mail servers.
- *Back-end archive repository* to store the archived e-mails: The repository can be Content Manager, OnDemand, or Tivoli Storage Manager. In this IBM Redbook, we focus only on Content Manager as the back-end repository.

You do not have to enable journaling or full-text indexing features for this type of the solution.

Implementing the solution for long-term retention and discovery

Over the years, e-mails have become a tool that people use to hold business correspondences and important documents such as contracts and financial spreadsheets. Some regulations require the company to retain all of the e-mails for a specific period of time. You must find a way to not only archive e-mails, but archive all the necessary e-mails, retain them for a specified time, and enable authorized personnel to search the e-mails from a central repository. That is, you have to implement an e-mail archive solution for longer-term retention and discovery.

Solution design: Defining software components

For a solution that is designed for long term e-mail retention and discovery, you have to integrate the following software components into your existing mail server environment:

- *CommonStore* to archive old e-mails to the back-end repository: Use CSLD or CSX based on the existing mail servers.
- *Content Manager* as the back-end archive repository to store the archived e-mails.
- *eMail Search component* that enables high performance full-text indexing of the archived e-mails for archiving and discovery purpose.

To archive all or specific sets of e-mails for discovery purpose, you have to enable the following features in your CommonStore solution:

- *Journaling* to ensure that all or a subset of e-mails are captured for archiving.
- *Full text search* to enable future discovery searches on the archived e-mails.

The only assured way to capture every single e-mail or e-mails from a specific user is through journaling. Enabling full-text indexing with eMail Search component ensures that you can full-text index large volume of e-mails, and ensure that the indexed e-mails are available to be searched for discovery purpose.

We recommend that you consult with IBM consultants to ensure that the solution design is feasible and optimal for your business requirement and environment.

3.3.2 Current mail system infrastructure and mail usage

To design a CommonStore solution, it is important to know what is the existing mail system infrastructure. For example, you must know the platforms that are used. You can install CommonStore on Windows® or AIX platforms. The operating system that you are using today for your mail servers most likely dictates the platforms that you use for the CommonStore solution.

To obtain the existing mail system infrastructure and usage, we include the following set of questions:

- How many mail servers do you currently have? Do you use clustering?
- If there are multiple mail servers, are they located at multiple sites? What is the topology and available bandwidth of the current infrastructure?

  The number of e-mails that your solution has to archive within a defined operation time determines the number of CommonStore servers that you require for your solution. Knowing the number of the mail servers you have and where they are located (and how they are being used) helps you to determine how to distribute the CommonStore servers across multiple mail servers.

  The *bandwidth of the network* might affect the archiving speed of the solution. This is especially true in a distributed environment where there are multiple mail servers located in geographically different locations.

  For example, if you implement a CommonStore solution for long-term retention and discovery and you have to centrally archive e-mails from different e-mail servers across different locations, then slow network speed affects how fast the e-mails can be archived to the central repository.

  In this type of the environment, assuming that you work with Domino server, it is possible to design a system such that you can:

  a. Journal the mail databases at the local mail servers.
  b. Replicate the journal databases from local servers to one central server.
  c. Archive e-mails from the replicated journal databases in the central server.

  This is where you might discover the potential bottleneck of your system. Work with a network engineer and test the network speed to ensure that the network throughput can accommodate expected file transferring rate between the servers. Design your system accordingly.

- On what operating systems are the mail servers running?

  For Domino server, it might be IBM z/OS®, AIX, IBM OS/400®, Linux®, Solaris™, or Windows. For Exchange server, it might be Windows NT®, Windows 2000 or 2003.

  Knowing this information along with the available platforms that CommonStore for Lotus Domino and Exchange Server support and the available platforms that Content Manager supports, helps you to determine the platform you use for each solution component. You must also take sizing into consideration. Make sure that the platform you use (along with the servers) can accommodate system performance requirement (that is, to be able to archive the expected number of e-mails within a specified period of time).

- What software do you use for backing up the mail servers?

  This can be Tivoli Storage Manager, EMC Legato Networker, Veritas NetBackup, or others.

  If you already use a specific system for backup, you can use the same system with Content Manager for long-term storage if it is compatible with Content Manager.

  For example, if the current system uses Tivoli Storage Manager for Domino server backup, it is only logical to configure Content Manager to use Tivoli Storage Manager in the e-mail archiving solution.

- What is the current disk capacity of all the mail servers together? How much of this storage is already consumed by the mail databases today? If you cluster your mail servers, what is the size of a single version of all the databases (not all the replica together)?

- What is the estimated rate at which the current storage is growing? If you cluster your servers, what do you estimate is the growth rate for a single replica of all the databases?

  The previous two questions give you an idea of the minimum storage size necessary to archive e-mails today and in future. It becomes critical when you want to implement a long-term e-mail retention and discovery solution where you have to journal e-mails and retain them for a specific number of years.

  The answers are used as input for sizing. We discuss sizing in Chapter 4, “Sizing” on page 45.

- How many e-mails are stored in an e-mail database on average per day? What is the average size of an e-mail excluding the attachment? What is the average number of recipients per e-mail?

You might not be able to get all the answers. If you cannot obtain the information, you can start with the following set of general principles based on some of the CommonStore systems that have been deployed in the past by the CommonStore lab services professionals:

- Typically, on average, a user might send or receive 10 to 20 e-mails per day.
- The size of an e-mail including its attachment usually ranges anywhere from 60 KB to 100 KB.
- On average, there are 2.5 recipients per e-mail including the sender.

If you are dealing with a solution for discovery purpose, and you have no idea how many e-mails users have every day, you can use the following assumption as a starting point:

- 40 unique e-mails per user per day

Use these numbers as a reference point. We recommend that you monitor your system to get a more accurate estimate of your mail system usage and pattern. The answers to these questions are the input used for sizing, which we discuss in Chapter 4, “Sizing” on page 45.

- What is the average size of an attachment? What is the average number of attachments per e-mail? What is the percentage of the e-mails that have attachments?

  Every part of e-mail, such as body, attachments, and attributes can be fully indexed. For a solution designed for the discovery purpose, all e-mails have to be indexed. It is important to know how many attachments will be indexed and what formats they have. This is because the attachment is converted to the text information by the INSO filter. For an attachment such as a PowerPoint® presentation, the size might be large, but the text within the presentation can be relatively small. For an attachment such as a Microsoft Word, the size of the document probably reflects the number of words that have to be indexed. The answers to these questions serve as input for estimating the database index size and other resource requirements, which we discuss in Chapter 4, “Sizing” on page 45.

- Has the Domino mail database archiving or Exchange Server PST file archiving been used for the mailboxes? If yes, are they archived on the local users’ PC or in the network? For Domino system, did you archive e-mails on a secondary Domino server? What is the estimated total size of these databases?

  This information is typically gathered by the mail administrator tallying each user’s PST file for a total amount of the disk space consumed. The implications are how much of this data has to be archived and how long it takes to archive the data.

- For Exchange Server, do you use Exchange Public Folders? If yes, do you use it for document sharing and shared mailbox? Have you developed custom forms and applications?

  This does not have any implication on the application design. However, if public folders are being used, you have to consider this for setting up policies.

- Is there a maximum size quota on users' mailboxes?

  When a quota is set on users' mailboxes, you need to know the potential implication. In some cases, quota can conflict with the archiving policy or can be misleading regarding the actual disk storage consumed by users' mail databases.

  For example, when there is no quota implemented, mailboxes grow, unless users decide to delete the e-mails. When you check the mail database size, it reflects more or less the actual size of users' mail databases.

  When quota is implemented, if users reach their quota, often the policy is set in place such that the users can no longer receive e-mails until they reduce their mailbox size to below the quota size. Users are forced to manually delete their e-mails regularly. A situation might arise such that when the system is ready to archive old e-mails, the e-mails are no longer there due to manual deletion by the user. On the contrary, if users expect automatic e-mail archiving and stop manually deleting the e-mails, the disk space consumed by users' mails might be bigger than originally calculated.

  When you design a solution, take the existing mailbox policy into consideration. If you have to implement a solution for long-term retention and discovery, pay particular attention to get as close an estimate as possible of the mailbox sizes when quota is used as part of the existing mailbox management strategy.

3.3.3 Specific business requirements

After you determine the primary purpose of the archiving solution and the current mail system infrastructure, you have to gather specific business requirements to help you design the solution. We break down the questionnaires into the following categories:

- Mailbox management specific questionnaires
- Long-term retention and discovery specific questionnaires
- Infrastructure requirements questionnaires

Mailbox management specific questionnaires

This set of questions is related to implementing a solution for mailbox management. If you have to implement a solution for long-term retention and discovery, move on to the following section, “Long-term retention and discovery specific questionnaires” on page 38.

When you design a mailbox management e-mail archiving solution, gather the following requirements:

- For how many users (in percentage) do you plan to introduce e-mail archiving for mailbox management?

  This number indicates how many users’ e-mails the system has to archive. It serves as an input for sizing calculation, which we discuss in Chapter 4, “Sizing” on page 45.

- How many hours are available to archive e-mails in your environment?

  The number of operation hours determines the number and the processing power of the CommonStore servers that you require. They also determine the processing power that other software components require. Use the value from this question as input for sizing. For more information, see Chapter 4, “Sizing” on page 45.

- How many users access their mail just in the office? What is the primary access method for the office users? For Domino system, do they use Notes client or IBM iNotes™? For Exchange system, what Outlook do they use (Outlook 2000, 2002 for XP, 2003, or Outlook Web Access (OWA))?

- How many users are traveling and require remote mail access? What is the primary access method for mobile users? For Domino system, do they use local replica of their mailbox, iNotes, or BlackBerry? For Exchange system, do they use Outlook Offline Access or OWA?

  The previous two questions help you to determine whether you have to implement mobile support for the solution.

- For Domino system, is your mail template customized? If yes, who did it? For Microsoft Outlook, are the Microsoft Outlook Forms customized? Every item that you open in Outlook uses a form to display information. Microsoft Outlook programming using Outlook Forms.

- Do you want to enable users to select specific e-mails and archive them manually?

  It is best to set up a CommonStore system to automatically archive e-mails based on an archiving policy rather than ask or allow users to manually do so. If there is such a requirement, always check why it is required and if doing so brings any real business value to the company.

- When do you want to remove archived e-mails from the system?

  This determines your deletion policy for the archived e-mails.

Long-term retention and discovery specific questionnaires

This set of questions is related to implementing a solution for long-term retention and discovery. If you have to implement a solution for mailbox management only, you can skip this section.

Gather the following information:

- For how many users do you plan to introduce the e-mail archiving solution?

  This number indicates how many users' e-mails the system has to archive. It serves as an input for sizing calculation, which we discuss in Chapter 4, "Sizing" on page 45.

- Which law, regulation, or policy do you have to comply with? Is it Securities and Exchange Commission (SEC), National Association of Securities Dealers (NASD), Department of Defense (DoD) 5015.2, Health Insurance Portability and Accountability Act (HIPAA), Sarbanes-Oxley Act (SOX), Food and Drug Administration (FDA) 21 CFR Part 11, or others? Are you required to capture all inbound and outbound e-mails?

  Some regulations require a company to capture every e-mail that passes through its mail server and store it in a separate database. This capture of e-mails is accomplished by journaling. Depending on the requirement, you can enable journaling on all users' mailboxes or on a special group of users. The storage requirement on the server increases due to journaling. There might be more requirement for the processing power also. For more information about how journaling works, see Chapter 6, "Journaling" on page 85.

  > **Tip:** To get an idea of how many e-mails pass through the users' mail database, you can turn journaling on for a set of mail databases for a day or two and obtain the number of e-mails that are journaled for the period. Use this number as the foundation to calculate the average number of e-mails a user receives per day.

  To provide discovery capability, you also have to install full-text search and eMail Search component, which leads to an increase in the database index space, and other hardware resource requirements. When sizing, you must take this into consideration. For more information about sizing, see Chapter 4, "Sizing" on page 45.

- How many hours are available to archive e-mails in your environment?

  The number of operation hours determines the number of CommonStore servers that you require and the processing power that you require for the servers. The answer to this question serves as an input for sizing. For more information, see Chapter 4, "Sizing" on page 45.

- How many old e-mails are archived initially after the CommonStore solution is implemented?

  If you want to implement a discovery solution and you have to capture and store a large volume of old e-mails, archiving them might impact the number and speed of the central processing units (CPUs) that you require for the solution and the storage size of the system (if they use the same system resources).

  For example, if you have to archive a million old e-mails with a single CPU server, it takes approximately 70 hours to complete the capture. If you plan to perform this activity at the same time that the system goes into production, then when you perform sizing, take into account the initial archiving load of the system. Estimate more CPUs for the server, if necessary. For sizing calculations, see Chapter 4, “Sizing” on page 45.

- Do you require e-mails to automatically expire (be deleted) after their retention period?

  This helps you to define the e-mail archiving and deletion policy.

> **Note:** In this IBM Redbook, we do not discuss integrating Records Manager with CommonStore. For information about the integrated solution, see *E-mail Archiving and Records Management Integrated Solution Guide Using IBM CommonStore and Records Manager*, SG24-6795.
>
> At the time of this writing, a new version of the Records Manager has been released. Although the referenced IBM Redbook uses an older version of Records Manager, many of the general concept and functions discussed in the IBM Redbook are still applicable.

Infrastructure requirements questionnaires

To design the system architecture, you have to figure out the infrastructure requirements. For proof of concept, you might not have the answers to all these questions. However, these are serious and important requirements to address. Gather or define these requirements at the beginning of the solution planning and designing phase, because this helps you to define the solution scope and avoid problems in future.

Gather the following requirements:

- What is the Recovery Time Objective for the solution?

  The Recovery Time Objective (RTO) determines the time that is required to rebuild the system after a disaster. For example, this can be seven days or less. The shorter the required time, the more complex the solution architecture.

- What is the Recovery Point Objective for the solution?

  The Recovery Point Objective (RPO) determines the amount of data that is lost during a disaster. For example, according to your business requirements, you can specify 24 hours to be the maximum data lost that your company can accept. If you take daily system backups, the RPO is 24 hours. The shorter the RPO requirement, the more complex the solution architecture.

- What is the required availability of the implemented system?

  This is usually measured in percent or hours of unplanned outages per year. For example, you can define it as 99% availability. This means that there is no unplanned downtime of the system of more than approximately 88 hours per year (out of 24 hours a day * 7 days a week * 52 weeks a year).

  For a solution that is designed for long-term retention and discovery, the important issue is to retain the e-mails and enable authorized personnel to perform search on the e-mails. The frequency to perform a search is usually rare or at a minimum level. Thus, the availability requirement for this type of solution should not be as high as a system that requires daily access by hundreds or thousands of users.

- What is your preferred platform for the solution, including hardware and operating system?
- What storage device do you plan to use for archived e-mails? Do you want to use Shark, EMC Symmetrix, DR550, or EMC Centera?

  The preferred platforms and storage device preference guide you in the sizing and designing of the e-mail archiving solution.

3.4 Sample CommonStore solutions

In this section, we present sample CommonStore solution architecture to help you to better understand how to design a CommonStore solution.

3.4.1 Sample solution 1

In this sample solution, we describe a company's business requirement and propose a system architecture for the solution. Use it as reference only.

Business requirement

The purpose of the company's initiative is to improve the efficiency of an e-mail system and enable cost reduction by delivering an e-mail management system.

The company wants to achieve the following objectives:

- Archival: Archive a large number of e-mails from the current Domino and Exchange servers to the tapes; eliminate the archival of duplicate e-mails.
- Local archive elimination: Remove the necessity for and the ability of users to create local archives.
- Discovery support: Provide a discovery model for litigation and audit support that can be completed by legal staff; remove the dependencies on IT support.
- Performance and scalability: The solution has to include a proven track record in archival records.
- Administration and operations: Must be intuitive for the system administrators.
- User experience: The user experience is to be intuitive with the goal of providing limited user training and learning curve.
- Support and training: Support must be provided 24x7 for all applications; formal training must be available for the administrative staff.
- Reduce total cost of storage: Allowing older documents to be stored on less expensive media without changing the user experience.
- Meet legal requests in a timely manner.
- Provide the legal department and administrators the ability to search e-mail throughout the enterprise without IT assistance; hence, a self-service application for legal that includes search and export.

Solution design

The sample solution environment includes Content Manager and eMail Search components on the same machine; the Domino server and the CommonStore product are installed on separate machines.

The sample architecture uses a three-tier approach:

- User clients tier
- E-mail server tier
- Content repository tier

Because this solution is designed for long-term retention and discovery, the e-mail journaling feature is enabled. A copy of all the e-mails that come into or are send out to the e-mail environment is captured in a journaled database, and archived into the Content Manager repository.

The copy of the e-mails that are routed to the users' inbox is managed using the company's e-mail policies. For example, archive all e-mails after 90 days.

After the e-mails are archived, they can be retrieved for viewing by the legal department or the system administrator upon request.

In the sample architecture, the legal department uses eMail Search client to conduct searches of the e-mails and perform discovery function if for a legal reason there is a need to prove the validity of the e-mails in question.

Figure 3-1 shows the logical system architecture diagram.

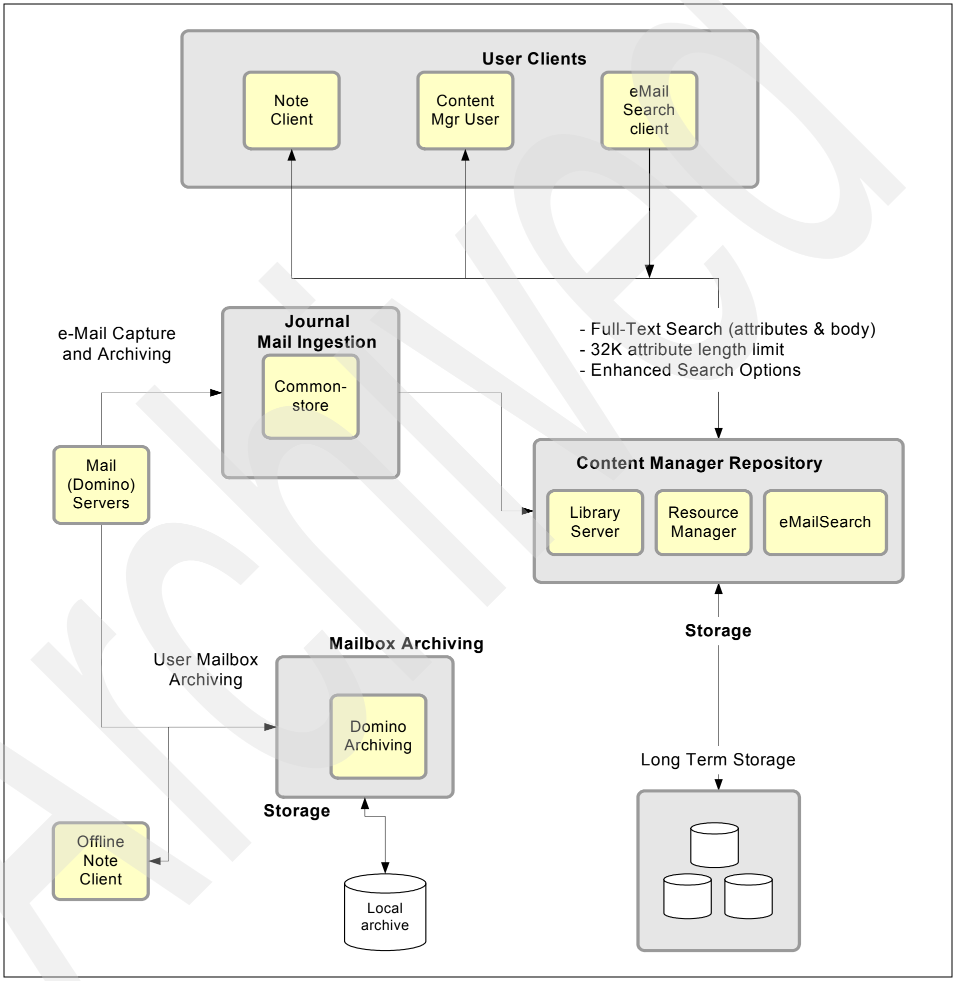


Figure 3-1 CommonStore for Lotus Domino sample system architecture

Figure 3-2 shows the physical system architecture diagram.

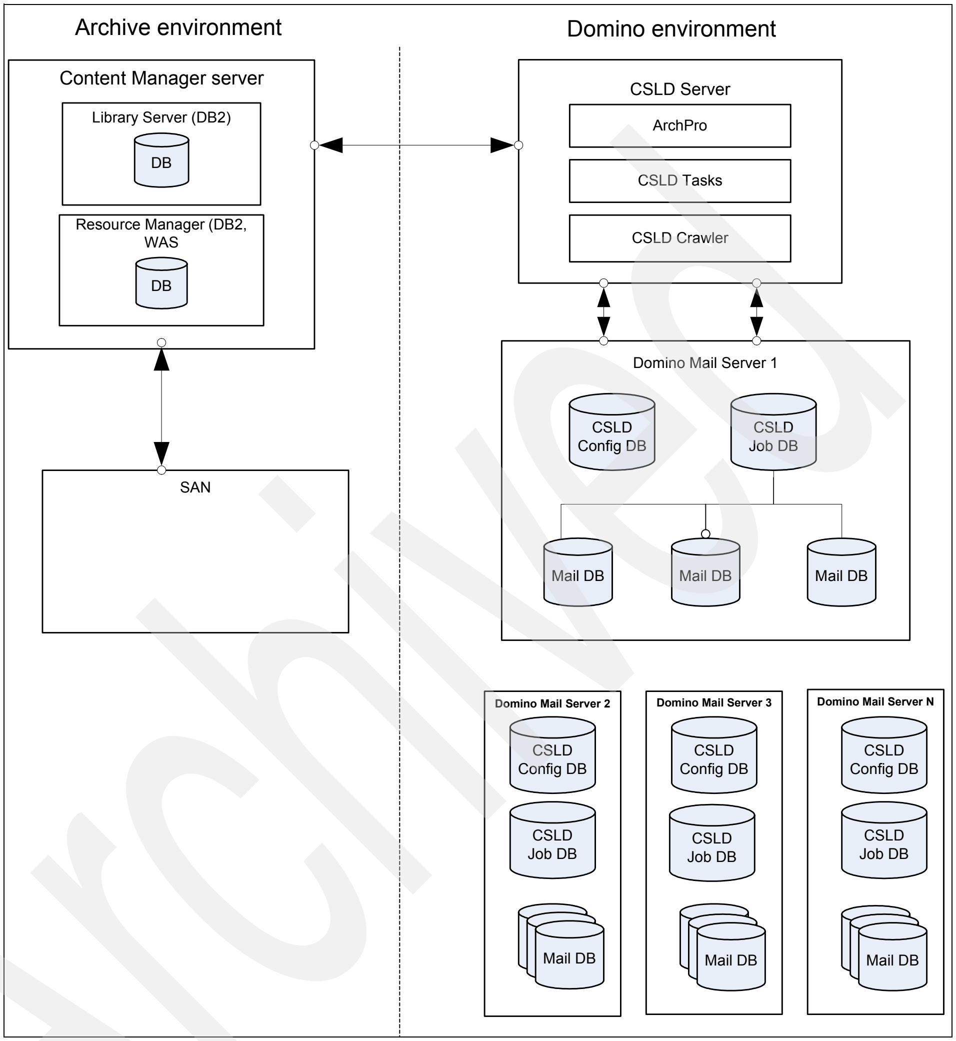


Figure 3-2 CommonStore for Lotus Domino solution architecture

3.4.2 Sample solution 2

Figure 3-3 shows a large system architecture sample. There are multiple CommonStore servers in the solution. Each CommonStore server serves multiple Microsoft Exchange servers. All e-mails are centrally archived into one Content Manager repository. The Content Manager Library Server and Resource Manager are on the same machine. Tivoli Storage Manager works with Content Manager for long-term e-mail storage.

Disaster recovery and high availability are currently not addressed in this design.

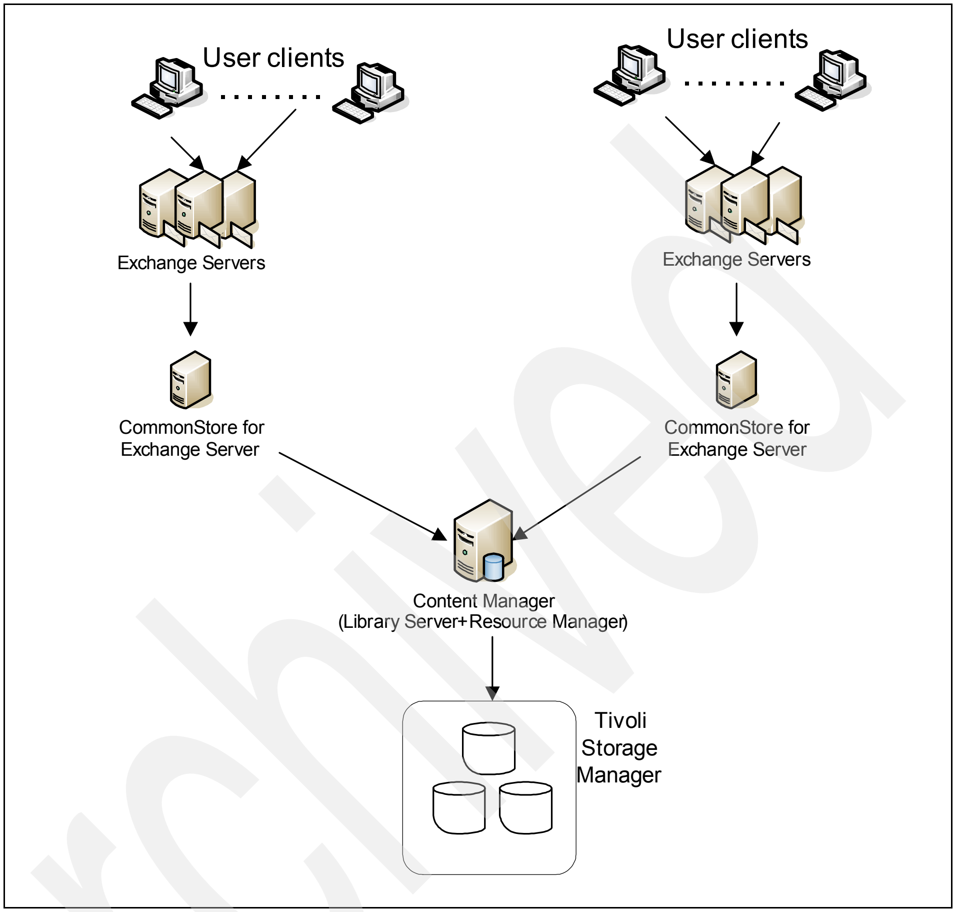


Figure 3-3 Sample large system architecture

This sample is provided as reference only. Design your solution based on current business infrastructure and business requirements. Work with IBM consultants when you design and finalize your solution design. We recommend that you implement the solution first in a nonproduction environment, such as in a test environment. This gives the flexibility of intensive testing and a managed roll-out of the solution.

4

Sizing

In this chapter, we discuss sizing for CommonStore solution implementations. There is a list of questions that you have to know related to sizing. We describe what they are and their impact on the CommonStore solution.

We cover the following topics in this chapter:

- Sizing concept
- Key information for CommonStore solution sizing
- Sizing questionnaires and the sizing implications
- Samples
  - Hardware estimation sample 1 (5,000 users, 20 e-mails per day)
  - Hardware estimation sample 2 (15,000 users, 20 e-mails per day)
  - Additional sizing samples
- Case study for eMail Search (discovery) sizing

4.1 Sizing concept

There are many reasons to implement a CommonStore archive solution. One common reason is to manage the size of the mailboxes in your organization. Another reason is for your organization to comply with certain regulations, such as retention of e-mail on Write Once, Read Many (WORM) storage, or to comply with a discovery order. Your solution has to accommodate hundreds, thousands, or hundreds of thousands of e-mails passing through the system every day. You might have to keep your e-mails for many years. Therefore, it is extremely important to ensure that the infrastructure that you put in place can accommodate archiving of the required volume of e-mails within a defined period of time and storing of these e-mails for a required length of period. You have to estimate the optimal hardware requirements. Is it possible to use the existing network and hardware? If not, what is the recommended hardware and network infrastructure to support the solution? To answer these questions and ensure the success of your solution implementation, you have to perform sizing.

A sizing is an approximation of the hardware resources required to support a specific software implementation. It is based on information available at a point in time and provides an entry to understanding the hardware requirements. Sizing does not replace capacity planning for the installed systems. When you are in the process of implementing your solution, work with an IBM Capacity Planning consultant to monitor and predict the ongoing resource requirements.

To perform sizing exercises, use the *sizing tool*. A sizing tool (also known as *sizer*) contains a set of calculations based on benchmark workloads developed and tested by the development lab in tuned systems in a lab environment. Using these numbers and the information that you provide, it estimates the hardware resources that is required to fulfill your business requirements. There are system and product configuration factors that are not taken into account. The sizer generally takes a conservative approach in the development of its estimates. The ultimate decision regarding the system must be made after a discussion about the entire implementation.

Before using the sizing tool, get a clear picture of the existing e-mail system infrastructure, its usage, and business requirement for the solution. We describe the set of questionnaires in Chapter 3, “Solution design” on page 27. The input from the questionnaires serves as input for the sizing tool to help you estimate the hardware resources required for your solution. Usually, you complete the questionnaires and work with an experienced IBM specialist to perform the sizing for your system. It is beneficial for you to understand how sizing is done. After reading through the chapter and working through the calculations, we hope that you have a basic understanding and use that knowledge to help in planning and allocating the right resources for your CommonStore solution in the beginning.

4.2 Key information for CommonStore solution sizing

The key information that you require to size a CommonStore solution are:

- The total number of unique e-mails that your system must be able to archive each day (within a specified time period).

  This information helps you to calculate the *number of central processing units (CPUs)* that is required for each software component and the *number of CommonStore servers* that you require (assuming that you use standard acceptable CPU speed for software components and CommonStore servers with standard processing power).

- Current average e-mail size (including attachments), the projected growing rate each year, and the retention policy of the archived e-mails.

  This information helps you to calculate the *disk space storage size* (used for the databases, objects, and index files) that you require today and for subsequent years.

When sizing for a CommonStore solution, we recommend the following best practices:

- Always calculate the disk space requirement over several years so that you have a better picture of the overall hardware requirement.
- Always get a machine that has at least certain standard CPU power and speed and still has the capacity to grow so that you can add more CPUs and disk spaces later when necessary. Do not size a machine that can accommodate only your first year CommonStore solution requirements. For example, if according to sizing, you only require an AIX machine with two CPUs for Content Manager, we recommend that you start with a standard AIX machine with four CPUs.
- If your company already uses the machines that have more processing powers, more memory, or more CPUs than the required machines from the sizing calculation, go with the same type of the machines as you currently have in the company if it is possible and feasible.
- When sizing a very large system for long-term retention and discovery, plan to spread data across multiple item types. eMail Search supports this feature. We recommend that you do not go more than 1 TB in an index. Check with the IBM lab consultants for the latest recommendation to handle large volume e-mails.

4.3 Sizing questionnaires and the sizing implications

The questionnaires that we presented in Chapter 3, “Solution design” on page 27, include the questions associated with sizing. We repeat the sizing-specific questions here for further discussion and their implication on sizing.

We group the questions based on two main area:

- Calculating CPU-related questionnaires
- Calculating storage space related questionnaires

4.3.1 CPU-related questionnaires and the sizing implication

As mentioned previously, to calculate the number of CPUs required for each software component and the number of CommonStore servers required, you have to estimate the total number of the unique e-mails that your system must be able to archive each day (within a specified time period).

This information might not always be readily available. To help the sizing tool to derive this information, the following set of questions are included in the questionnaires:

- How many users or mailboxes will use the solution (or percentage of user population who will use the solution)?
- How many e-mails are received or sent per user per day?

  By multiplying the two numbers from these questions, you can derive the total number of e-mails that the system must be able to archive each day. The answer to the second question might be hard to obtain. There is no set way to do this. We recommend that you monitor your existing system to get an estimate.

  If you are implementing a long-term retention and discovery solution, journal and archive all e-mails or a group of e-mails. We recommend (if feasible) that you turn on journaling for the targeted mail databases for a day or longer to observe the average numbers of message that are journaled.

  If you are implementing a mailbox management solution, you also have to figure out how many e-mails from the users’ databases your system must be able to archive. Take a survey of how many e-mails are in the current database as a starting point, if possible. Take into account that users delete their e-mails and that this might change after the solution is implemented. For example, if there is a quota on user mailboxes’ maximum size, users are forced to delete their e-mails regularly to maintain the quota. When they know that the new system can archive their e-mail automatically, they might not delete their e-mails manually anymore. In consequence, the system archives more e-mails per day than originally planned. Estimating the number for today might not accurately represent the number when the system is implemented. When sizing, estimate sufficient processing power to complete the e-mail archives.

- What is the average number of recipients plus sender within the same e-mail domain?

  If a user sends an e-mail to multiple people within the same mail domain, the system can store just one copy (instance) of the e-mail. For example, if person A sends an e-mail to person B and they both belong to your company, then there is a total of two e-mails among them even though there is only one version of the e-mail. In this case, you can set up the system to be smart enough to store only one instance of the e-mail. In other words, the CommonStore system only has to archive one unique e-mail instead of two. This affects the number of e-mails that the system archives every day.

  The total number of unique e-mails that the system archives is equal to the total number of e-mails divided by the average number of recipients plus the sender within the same e-mail domain.

Note: If you do not have an idea about the number of e-mails that the system has to archive a day, you can start with the following assumptions:

- Typically, for a mailbox management solution, it is approximately 10 to 20 e-mails per day per user.
- For a solution designed for long-term retention and discovery where you have to keep all the e-mails (or a group of e-mails), you can use 40 unique e-mails per day per user.

- What is the growth in the number of e-mails or messages per year?

  This helps you to estimate the future growth in the total unique e-mails that the system must be able to archive.

- How many hours are available to archive e-mails in your environment?

  This helps you to estimate the total number of CommonStore servers and the number of CPUs, and the different software components required for your solution. See “Sizing implication” for more details.

- How many old e-mails have to be processed as part of the initial system loading?

  If you size some hardware that requires nearly all the resources to handle the production load, and you have to process old e-mails at the same time when the system goes into production, consider increasing the hardware required to accommodate the extra workload in the beginning.

Sizing implication

Factors that affect system performance include: Disk input/output (I/O), CPUs of the servers, memory size, and others. According to benchmark testing, we can roughly estimate that CommonStore for Lotus Domino server can archive four e-mails per second per server; CommonStore for Exchange Server can archive approximately three e-mails per server. This assumes four parallel processes (CommonStore agents) with decent CPU speed.

Given the total number of unique e-mails that the system must archive, and the total number of hours that the system has to archive every day, you can estimate the number of CommonStore servers that you require to complete the archiving activity each day.

In addition, you can also calculate the CPUs required for other software components, for example, the Content Manager Library Server and Resource Manager. See 4.5, “Case study for eMail Search (discovery) sizing” on page 58, for details about how to derive the number of CPUs necessary for Library Server and Resource Manager.

> **Note:** Although a CommonStore solution can generally archive four e-mails (in Domino environment) and three e-mails (in Exchange environment) per CommonStore server, there is a limit to the maximum number of e-mails that you can archive when dealing with a solution that is designed for long-term retention and discovery.
>
> This limitation is due to the Net Search Extender (NSE) indexing speed. More CPUs do not change how many e-mails the system can index, but faster CPUs do affect this number. At the time of writing, we can safely assume that with optimal configuration, a system can index 25 e-mails per second with one Content Manager item type.
>
> Let us assume that the system can archive and index 25 e-mails per second, we also assume 40 e-mails per user per day, a 16-hour processing day (for e-mail archiving and indexing), with an 8-way machine, you can support 36,000 mailboxes per day, or 1.44 million e-mails per day.

4.3.2 Storage space related questionnaires and the sizing implication

As mentioned in 4.2, “Key information for CommonStore solution sizing” on page 47, to calculate the number of storage space for databases, index files, and objects, you have to obtain the current e-mail size, the projected growth rate each year, and the retention policy of the archived e-mails.

This information might not always be readily available. To help the sizing tool to derive this information, the following set of questions are included in the questionnaires:

- What is the average e-mail text body size?
- What is the average e-mail attachment file size?
- What is the average number of attachments per e-mail?
- What percentage of e-mails include attachments?

  These four questions help you to calculate the average e-mail size. Given the total number of unique e-mails to be archived per day (see 4.3.1, “CPU-related questionnaires and the sizing implication” on page 48) and the average size of e-mail from this section, you can calculate the space required to store all the e-mails for the current year. For calculation formulas, see 4.5, “Case study for eMail Search (discovery) sizing” on page 58. Be aware that sometimes the system compresses the attachments. For example, Domino always try to compress attachments. Therefore, the sizes might be different than the actual size of attachments.

- What is the growth in average size of e-mails per year?

  This helps you to calculate future growth in storage space usage.

- What is the retention periods for e-mails?

  You should determine the retention period and the percentage of e-mails with this retention period. For example, you can specify that you want to keep 50% of the e-mails for three years, and the other 50% for five years. Providing this input to the sizing tool, you can calculate the storage space that is required for the e-mails for the year.

- What is the number of days that the objects must be kept on disk managed by the Resource Manager before being migrated to external storage device?

  Assuming that you use Content Manager as the archived repository, this determines how much hard disk space on Resource Manager you require for storing objects. The minimum required size on the server can be roughly calculated as the percentage of e-mails with an average e-mail size multiplied by the number of days that you keep them on the server before migrating them to the external storage device.

- Will the Resource Manager be replicated?

  If you want to replicate the Resource Manager, estimate the additional hardware requirements on the replicated server to keep the same objects as on the primary server and the additional CPUs on the primary server to manage the calls from the replicated Resource Manager to stored procedure on the Library Server to change the status of the request from pending to complete.

Sizing implication

The set of the questions provided in this section affects the overall storage requirement for your solution.

After you know the retention period of e-mails in archive, you can calculate the minimal storage size that is required for the retention period.

For example, let us assume that you want to implement a discovery solution and you must keep e-mails for five years. The number of unique e-mails to be stored per day is 10,000 with an average size of 80 KB. Full text feature is enabled.

Based on this number, you have (assuming 5 working days per week):

```
10,000 e-mails * 52 weeks * 5 days = 2.6 million e-mails
```

In other words, you have to archive 2.6 million e-mails per year.

Content Manager creates:

- One row in the Library Server’s root component table per e-mail stored
- One row in the Library Server’s child component table per e-mail stored
- One row in the Resource Manager’s database table per e-mail stored

When archiving 2.6 millions of e-mails, this translates to:

- 2.6 millions of rows in the Library Server’s root component table
- 2.6 million of rows in the Library Server’s child component table
- 2.6 million of rows in the Resource Manager’s database table

We use the following constant:

- For the Library Server database, each row in the root component table takes 1800 bytes, and each row in the child component table takes 300 bytes.
- For the Resource Manager database, each row takes 350 bytes per object.

Based on this, the calculated Library Server database space required to store e-mails is:

```
2.6 million rows * 1800 bytes + 2.6 million rows * 300 bytes = 5.1 GB
```

In other words, you require 5.1 GB of Library Server database space for a year.

The calculated Resource Manager database space is:

```
2.6 million rows * 350 bytes = 0.8 GB
```

The calculated Resource Manager object store size is:

```
2.6 million e-mails * 80 KB * object growth factor (1.2 * 1000) = 232.5 GB
```

For more detailed information about how to determine these numbers and the object growth factor value, see 4.5, “Case study for eMail Search (discovery) sizing” on page 58.

> **Important:** When sizing a very large system for long-term retention and discovery, plan to spread data across multiple item types. The NLV of eMail Search supports this feature. We recommend that you do not go more than 1 TB in an index. Check with the IBM lab consultants for the latest recommendation to handle large volume e-mails.

4.4 Samples

In this section, we provide sizing samples output and discuss how to derive some of the calculations.

If you plan to implement a CommonStore solution for discovery and want to know the details of how the calculations are derived, go directly to 4.5, “Case study for eMail Search (discovery) sizing” on page 58.

Figure 4-1 shows the assumed system architecture for a sample CommonStore solution.

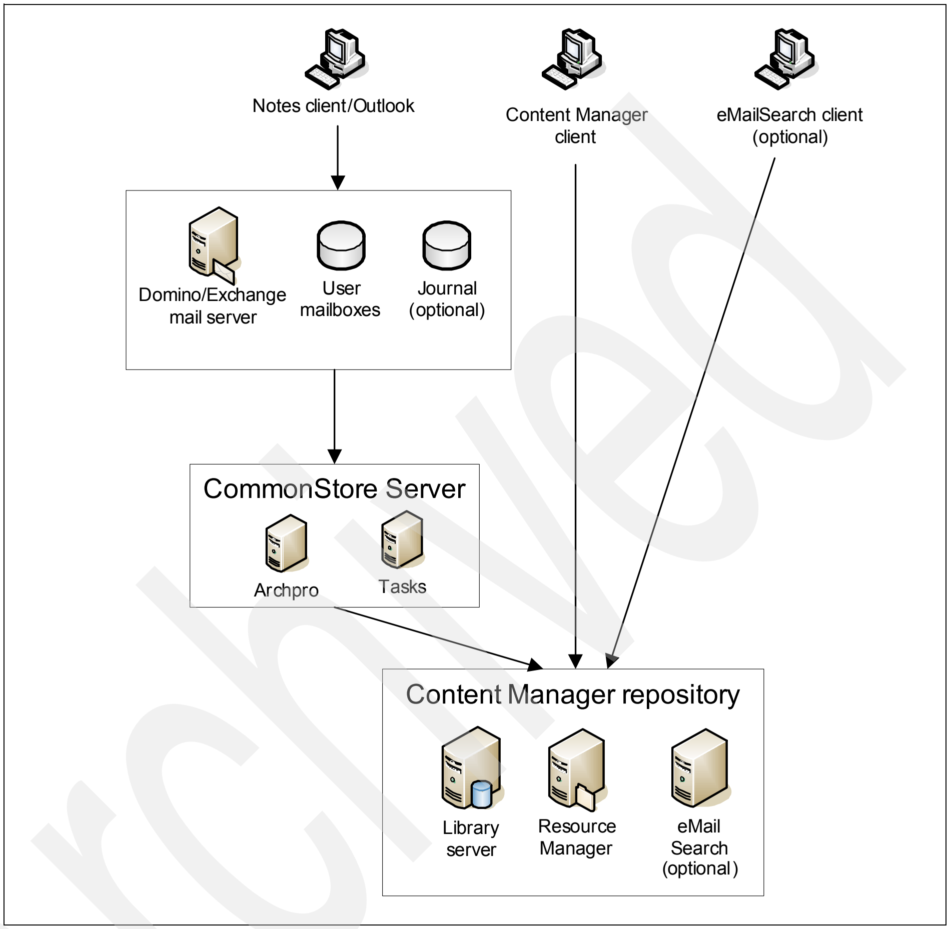


Figure 4-1 CommonStore sample architecture

4.4.1 Hardware estimation sample 1 (5,000 users, 20 e-mails per day)

Let us consider the first sample hardware estimation that handles 100,000 e-mails archiving per day.

Assumptions:

- Number of users: 5,000
- Average number of e-mails per user per day: 20
- Average e-mail message size: 80 KB

- Average number of recipients plus sender: 2.5
- Average size of e-mail attachment: 100 KB
- Average number of e-mail attachments: 1
- Number of available hours for CommonStore to run: 8
- 100 MBit connection between mail servers, CommonStore, and Content Manager
- Content Manager server is configured to handle the volumes sent by CommonStore

Calculations:

```
5,000 users * 20 e-mails per day = 100,000 e-mails
```

This (100,000) is the number of e-mails that your system must be able to archive within the 8 hours of operation. We defer the rest of the sizing calculation to the later sections.

Hardware requirements:

To implement a mailbox management solution, the estimated hardware requirements are:

- 1 Content Manager server with 4 CPUs, 8 GB RAM (AIX)
- 1 CommonStore server with 4 CPUs, 4 GB RAM (AIX)

When introducing journaling and eMail Search to provide long-term e-mail retention and discovery solution, the estimated hardware requirements are:

- 1 Content Manager server with 4 CPUs, 10 GB RAM (AIX)
- 1 CommonStore server with 4 CPUs, 4 GB RAM (AIX)

Tip: Although the calculation shows that you require a CommonStore server and a Content Manager server with 2 CPUs each, we recommend that you go with the servers with 4 CPUs each.

4.4.2 Hardware estimation sample 2 (15,000 users, 20 e-mails per day)

Let us consider the second sample hardware estimation that handles 300,000 e-mails archiving per day and how this number affects the number of CommonStore servers required for the solution.

Assumptions:

- Number of users: 15,000
- Average number of e-mails per user per day: 20
- Number of available hours for CommonStore to run: 8

The rest of the assumptions are the same as the previous sample.

Calculations:

```
15,000 users * 20 e-mails = 300,000 e-mails
```

This (300,000) is the number of e-mails your system must be able to archive within the 8 hours of operation.

```
300,000 e-mails / 8 hours / 3600 seconds = 10.42 e-mails/second
```

This (10.42) is the number of e-mails that the system must be able to archive per second. To calculate the number of Intel/AMD servers, assuming that we use four parallel processes, we derive the total number of CommonStore servers that we require.

```
10.42 / 4 = 2.6 or 3 CommonStore servers
```

Hardware requirements:

On IBM System p5™ server running on AIX with fast I/O disk subsystem, it is estimated that CommonStore can archive as much as 30,000 e-mails per hour, running six to eight parallel processes.

Therefore, 10.42 / 6 parallel processes = 1.74 or two CommonStore Servers. By using faster I/O disk subsystem, we can conclude that the number of CommonStore servers required to complete the task can be reduced by 1/3. As shown previously, it requires three CommonStore Servers running on Intel/AMD to complete the same task as two CommonStore Servers running on System p5 server running on AIX.

4.4.3 Additional sizing samples

Assuming that we are sizing for a CommonStore mailbox management solution, we present the sample sizing output based on a set of the input values according to the questionnaires.

Table 4-1 shows the responses for the questionnaires. These responses serve as input for the sizing tool calculation.

Table 4-1 E-mail questionnaire

Questions	Responses
Average number of e-mails per user per day	20
Average size of an e-mail including attachments (KB)	100 KB
Average number of recipients per e-mail	2.5

Also used in the calculation is the assumption that there are five working days per week and we use 52 weeks per year.

Table 4-2 shows the hardware recommendations according to the output from the sizing tool. The recommendation is grouped by the total e-mails that have to be archived per day for the solution. For convenience, we also include the number of users for each group.

Table 4-2 Recommended hardware requirements

e-mails per day (users)	20 - 2,000 (1 - 100)	2,000 - 20,000 (100 - 1,000)	20,000 - 100,000 (1,000 - 5,000)	100,000 + (5,000+)
Server	IBM eServer™ xSeries® 346, 4 GB p5-505, 4 GB	xSeries 346, 4 GB p5-505, 4 GB	p5-520 4-way 8 GB x3850 or x366 4-way 8 GB	2 servers or LPARs: 8-way 16 GB for CM 4-way 8 GB for CSLD
Initial fast direct access storage device (DASD)	100 GB	100 GB	300 GB	1 TB
Storage (GB) per year (Storage per user per year * number of users)	24.8	495.9	2479.6	9918.2
3-year storage requirements (GB)	74.4	1487.7	7438.7	29754.6
DR550 usage storage (TB)	1.0	3.4	4.8	9.6

4.5 Case study for eMail Search (discovery) sizing

In this section, we use the questions in 3.3, "Requirement-gathering questionnaires" on page 30, the sizing tool, and provide a complete and detailed example of eMail Search sizing questionnaires, responses, and the resulting calculations.

4.5.1 Calculating CPUs required for the solution

Based on the previous 15,000 users, 20 e-mails per user example, in Table 4-3, we provide a completed sizing questions and responses for the solution requirement. This information is used as input for the sizing calculation tool. Note that not all the questions presented in 3.3, "Requirement-gathering questionnaires" on page 30 are listed here. This is because only this set of the questions is used as input values for the sizing tool. The rest of the questions that we covered previously are there to help you get a complete picture of your current infrastructure, your business requirements, and why you want to implement the solution. A complete understanding of what you have and what you want to accomplish helps you to fine-tune the resource estimates that you get from the sizing tool.

Table 4-3 eMail Search sizing questionnaire

Questions	Responses
Number of users or mailboxes	15,000
Number of e-mails per user per day	20
Percent of growth in the average size of e-mails	10%
Percent of growth in the number of e-mails	10%
Average number of recipients plus sender within the same mail domain	2.5
Average size of the e-mail body or message text (thousand bytes)	80.0
Average size of e-mail attachments (thousand bytes)	100.0
Average number of attachments	1
Percentage of e-mail with attachments	50%
How many old e-mails can be processed as part of an initial loading	0
What retention periods are required, and what percent of e-mail can be kept for each retention period? ▶ 3 years	100%
Number of hours to complete capture (archive)	8

Questions	Responses
How many days can objects be kept on hard disks managed by the Resource Manager before they are migrated to the external storage?	0
What retention managed storage device with Content Manager can be used?	Tivoli Storage Manager
What percent of e-mail can be stored on retention managed storage versus other media?	100%
Will Content Manager replication be used?	No

In the calculation, we assume that there are 24 working days per month for all the mail users.

The output is calculated for five years and it includes the capacity requirements and hard disk requirements. Capacity calculation takes into account the growth of e-mails every year, the number of hours that the CommonStore server can run (CommonStore agents can work), usage of Single-Instance Storage (SIS) to store only one copy of distributed e-mail, and growth of e-mail size and volume per year. Table 4-4 shows the result.

Table 4-4 Calculated capacity requirements

Capacity requirements:	**Year 1**	**Year 2**	**Year 3**	**Year 4**	**Year 5**
E-mails per day	300,000	330,000	363,000	399,300	439,230
Hours per day (CommonStore)	8	8	8	8	8
E-mails per second (CommonStore)	10.42	11.46	12.60	13.86	15.25
E-mails per day after SIS	120,000	132,000	145,200	159,720	175,692
E-mails per second after SIS	4.17	4.58	5.04	5.55	6.10
Average size of e-mail (KB)	130.0	143.0	157.3	173.0	190.3
Data rate to Resource Manager (MB/sec)	1.32	1.60	1.94	2.34	2.83
When using System p5 1.65 GHz CPUs required:					
► Library Server CPUs (raw)	0.08	0.1	0.1	0.1	0.1
► CPUs required for indexing	1.0	1.1	1.3	1.4	1.5
► CPUs required for SIS search	0.1	0.1	0.1	0.1	0.2

Capacity requirements:	Year 1	Year 2	Year 3	Year 4	Year 5
► Total Library Server CPUs	1.2	1.4	1.5	1.6	1.8
Library Server CPUs (store, search, retrieve, rounded)	2.0	2.0	2.0	2.0	3.0
► Resource Manager CPUs (raw)	0.3	0.3	0.3	0.3	0.4
► Resource Manager CPUs (store, estimate for retrieve, migrate)	0.4	0.4	0.5	0.5	0.5
Resource Manager CPUs (rounded)	1.0	1.0	1.0	1.0	1.0
Number of CPUs for CSLD (maximum 4-way machines)[a]	3	3	4	4	4

[a] In case you are using IBM eServer pSeries®, the Library Server and Resource Manager should be on the same logical partition (LPAR) unless AIX 5.3 micropartitioning is used. In case you are using Windows platform, we assume high-end Windows CPU is 75% of pSeries. This has not been validated. It is only an assumption that we use for the calculation here.

This example calculates the number of CPUs required for Library Server, Resource Manager, and CommonStore server.

Deriving calculated CPU requirements for year one

Along with this IBM Redbook, we provide the sizing tool that produces the results as shown in Table 4-4. You can download the sizing tool from the IBM Redbooks site. For details about how to download the tool, see Appendix B, “Additional material” on page 201.

Disclaimer: Use the sizing tool that we provide with this IBM Redbook and the calculation details that we describe here to understand how sizing can be done. Do not use them without consulting the IBM specialists in this area. Always check with the Content Management lab for the most up-to-date sizer if you are eligible to obtain one. The lab continues to gather data to refine these calculations. Use the formulas that we provide here to get an idea of how sizing work. Do *not* use the sizing tool as is.

To help you better understand the sizing tool, we provide some of the calculations shown in Table 4-4 according to the sizing tool formulas. We recommend that you download the sizing tool and go through each calculation or have a spreadsheet ready to do the calculation. At a minimum, we recommend that you print out Table 4-3 on page 58 and Table 4-4 before you go through the calculation details.

Some of the calculations have been mentioned in the previous section. We include them here for the completeness of the discussion and easy reference.

E-mails per day: This is the total number of e-mails that your target solution must be able to archive per day. Based on the number of users who use the system and the average number of e-mails per user per day, you can calculate this figure according to Figure 4-2.

```
E-mails per day
= Number of users or mailboxes * Number of e-mails per user per day
= 15,000 * 20
= 300,000 e-mails need to be archived by the system per day
```

Figure 4-2 Formula to calculate total number of e-mails to archive per day

Hours per day (CommonStore): This is the total number of hours that your system can archive per day. This figure is equal to the response value from the questionnaire, the number of hours to complete the capture (archive). See Figure 4-3.

```
Hours per day (CommonStore)
= Number of hours to complete capture (archive)
= 8 hours a CommonStore system has to archive e-mails per day
```

Figure 4-3 Formula to calculate hours per day

Note: Typically, you have up to 20 hours to capture e-mails. Even if the archiving (capture) time is limited to 8 hours for CommonStore server, the Content Manager server (as the back-end repository) can create a full-text index during 20 hours.

The more the hours available to archive e-mails, the less hardware required to be put in place.

E-mails per second (CommonStore): This is the total number of e-mails that the CommonStore server must archive per second. This number is derived from the total number of e-mails that you must archive per day and the total number of the hours that your system has to archive e-mails each day. See Figure 4-4.

```
E-mails per second (CommonStore)
= E-mails per day / Hours per day / total seconds per hour
= 300,000 / 8 / 3600
= 10.42 e-mails per second
```

Figure 4-4   Formula for e-mails per second the CommonStore server has to archive

E-mails per day after SIS: This is the number of unique e-mails per day that the system must archive if you enable the SIS feature. This number is based on the total e-mails to be archived and the number of recipient plus sender per e-mail within the same e-mail domain. See Figure 4-5.

```
E-mails per day after Single Instance Store
= E-mails per day /
  Average number of recipients per e-mail plus sender
  within the same mail domain
= 300,000 / 2.5
= 120,000 e-mails per day after enabling Single Instance
```

Figure 4-5   Formula to calculate e-mails per day after enabling SIS

E-mails per second after SIS: This is the number of e-mails that the system must archive after SIS is enabled. This number is based on the e-mails per second without SIS consideration and the number of recipient plus sender per e-mail within the same e-mail domain. See Figure 4-6.

```
E-mails per second after SIS
= E-mails per second /
  Average number of recipients per e-mail plus sender
  within the same mail domain
= 10.42 / 2.5
= 4.17 e-mails per second after SIS
```

Figure 4-6   Formula to calculate e-mails per second for archiving with SIS

> **Note:** Although we show how to calculate the e-mails per day and per second values after enabling the SIS feature, for some sizing calculations, we do not take this feature into consideration.

Average size e-mail (KB): This is the average size of an e-mail. This number is based on the average e-mail body size, the average e-mail attachment size, the average number of attachments per e-mail, and the percentage of the e-mails that has attachments. See Figure 4-7.

```
Average size e-mail (KB)
= Average size of e-mail body or message text (thousand bytes) +
  (Average size of e-mail attachments (thousand bytes) *
  Average number of attachments *
  Percentage of e-mail with attachments)
= 80 + (100 * 1 * 50%)
= 130 (KB)
```

Figure 4-7 Calculating the average e-mail size including the size of the attachments

Data rate to Resource Manager (MB/sec): This is the data rate that the Resource Manager has to handle per second. This number is based on the total e-mails per second that the CommonStore system has to handle and the average size of an e-mail. See Figure 4-8.

```
Data rate to Resource Manager (MB/sec)
= E-mails per second (CommonStore) *
  Average size e-mail (KB) / number of KB per MB
= 10.47 * 130 / 1024
= 1.32 (MB/sec)
```

Figure 4-8 Calculating data rate Resource Manager has to handle

> **Note:** The following Library Server and Resource Manager calculations are based on System p5 with a CPU speed of 1.65 GHz.
>
> As mentioned previously, these calculations are based on previous client experience and lab benchmark testing. Always check with the Content Management lab for the most up-to-date sizer if you are eligible to obtain one. The lab continues to gather data and refine these calculations. You should use the formulas we provide here to get an idea of how sizing work. Do *not* use the sizing tool and its calculation as is.

Library Server CPUs (raw): These are the raw CPUs required by the Library Server to process the e-mails. It is based on Library Server CPUs required per e-mail per second, the CPU speed ratio, and the number of e-mails per second that the system has to archive. Note that it is beyond the scope of this IBM Redbook to cover the details of how these numbers are obtained. In the sizing calculation, we use the values based on the sizing tool. For the calculation formula, see Figure 4-9.

Library Server CPUs required for indexing: These are the CPUs required by the Library Server to perform indexing of the data. This is based on the indexing rate per CPU and the CPU speed ratio. As mentioned previously, besides providing the numbers for these values and using them in the calculation, we do not provide details about how these numbers are derived in this IBM Redbook. For the calculation formula, see Figure 4-9.

Library Server CPUs required for SIS Search: These are the CPUs required by the Library Server to process searches required for the SIS feature. This number is based on the number of recipients per e-mail plus sender within the same mail domain. For the calculation formula, see Figure 4-9.

Total Library Server CPUs: This is a summary of the previous three values, Library Server CPUs (raw), Library Server CPUs required for indexing, and Library Server CPUs required for the SIS search. Figure 4-9 shows the calculation.

Library Server CPUs (store, search, retrieve, rounded): This is the rounded up value that the total Library Server CPUs estimated. Figure 4-9 shows the calculation.

```
Library Server CPUs (raw)
= Library Server CPUs per e-mail per second (Does not include SIS) /
  CPU Speed Ratio (1 = 1.65 GHz p5) *
  E-mails per second after Single Instance Store
= 0.020 / 1.00 * 4.17
= 0.08

Library Server CPUs required for Indexing
= E-mails per second after Single Instance Store /
  (Indexing rate (Documents per second per CPU) *
  CPU Speed Ratio (1 = 1.65 GHz p5))
= 4.17 / (4 * 1.00)
= 1.0

Library Server CPUs required for SIS Search
= E-mails per Second CommonStore /
  E-mails per second after Single Instance Store *
  Library Server CPUs (raw) * 0.5
= 10.42 / 4.17 * 0.08 * 0.5
= 0.10

Total Library Server CPUs
= Library Server CPUs (raw) +
  Library Server CPUs required for Indexing +
  Library Server CPUs required for SIS Search
= 0.08 + 1.0 + 0.10
= 1.2

Library Server CPUs (store, search, retrieve, rounded)
= Round up to integer (Total Library Server CPUs + 0.25)
= Round up to integer (1.2 + 0.25)
= 2.0
```

Figure 4-9 Calculating CPUs the Library Server has to store, search, and retrieve

Resource Manager CPUs (raw): These are the raw CPUs that the Resource Manager requires to process the e-mails. It is based on the Resource Manager CPUs required per e-mail per second, the CPU speed ratio, and the number of e-mails per second that the system has to archive.

Resource Manager CPUs (store, estimate for retrieve, migrate): These are the estimated CPUs the Resource Manager requires based on the raw CPUs required.

Resource Manager CPUs (rounded): This is the rounded up number of CPUs required by the Resource Manager as calculated previously.

In Figure 4-10, we show the calculation for the Resource Manager related values.

```
Resource Manager CPUs (raw)
= Resource Manager CPUs per e-mail per second (Exclude SIS) /
  CPU Speed Ratio (1 = 1.65 GHz p5) *
  E-mails per second after Single Instance Store
= 0.060 / 1.00 * 4.17
= 0.3 (round off, actually 0.25)

Resource Manager CPUs (store, estimate for retrieve, migrate)
= Resource Manager CPUs (raw) * 1.5
= 0.25 * 1.5
= 0.4

Resource Manager CPUs (rounded)
= Round up to integer (Resource Manager CPUs + 0.25)
= Round up to integer (0.4 + 0.25)
= 1.0
```

Figure 4-10 Calculating the CPUs the Resource Manager needs to store, search, and retrieve

Number of CPUs for CSLD (maximum 4-way machines): This is the estimated number of CPUs required to archive e-mails by the CommonStore system. It is based on the assumption that with an optimal system using a standard minimum of CPU, you can process approximately four e-mails per second. See Figure 4-11.

```
Number of CPUs for CSLD (maximum 4-way machines)
= Round up to integer
  (E-mails per Second CommonStore /
  E-mails per second per CPU for CSLD)
= Round up to integer (10.42 / 4)
= 3.0
```

Figure 4-11 Calculating the number of CPUs required by the CommonStore server

Summary of the calculated results for year one

Based on the previous calculation, to handle e-mail archiving for year one, you require:

- A System p5 with 1.65 GHz CPU or later, and a minimum of two CPUs for the Library Server and one CPU for the Resource Manager
- A CommonStore for Lotus Domino server with three CPUs, or multiple servers, for example, three 4-way servers

> **Important:** These calculations are provided here to help you understand the sizing process. Always check with the Content Management lab for the most up-to-date sizer if you are eligible to obtain one. The lab continues to gather data and refine these calculations. Do *not* use the provided sizing tool without consulting the IBM sizing professionals.

Deriving calculated results for the next four years

Based on the responses from the questionnaires, the growth in average size of e-mails and e-mail volume is 10% each for the following year. The calculation for future years are based on the numbers from the previous years.

E-mails per day: This number increases by 10% each year. Figure 4-12 shows the formula and example of calculating the number for year two and year three.

```
E-mails per day for year two
= E-mails per day (in previous year) * (1 + % growth)
= 300,000 * (1 + 10%)
= 330,000 e-mails to be archived per day for year two
E-mails per day for year three
= E-mails per day (in previous year) * (1 + % growth)
= 330,000 * (1 + 10%)
= 363,000 e-mails to be archived per day for year three
```

Figure 4-12 Calculating e-mail per day for year two and year three

Average size e-mail (KB): This number increases by 10%. Figure 4-13 shows the formula and the example of calculating the number for year two and year three based on the number from the previous year.

```
Average size e-mail (KB) for year two
= Average size e-mail (in previous year) * (1 + % growth)
= 130 * (1 + 10%)
= 143 KB for average e-mail size for year two
Average size e-mail (KB) for year three
= Average size e-mail (in previous year) * (1 + % growth)
= 143 * (1 + 10%)
= 157.3 KB for average e-mail size for year three
```

Figure 4-13 Calculating the average e-mail size for year two and year three

Figure 4-14 shows the CPU requirements for CommonStore for Domino servers, Library Server, and Resource Manager according to the sizing calculation.

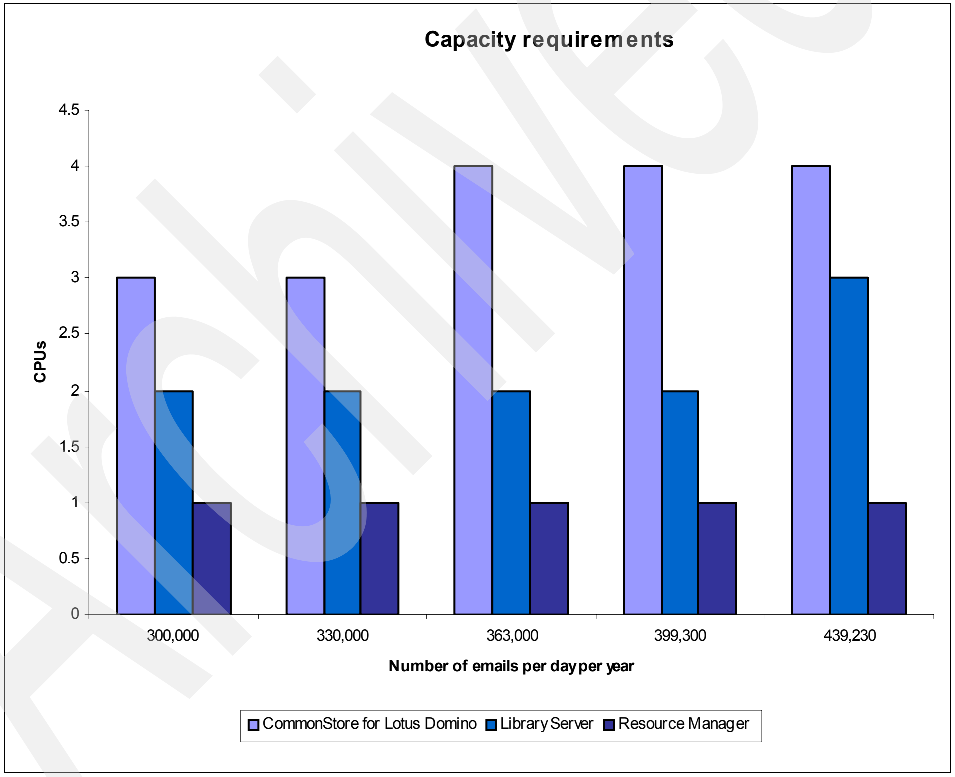


Figure 4-14 Capacity requirements for CPU per number of e-mails per day per year

4.5.2 Calculating storage space needed for the solution

Continuing with the previous example, with 15,000 users and an average of 20 e-mails per user per day, we estimate the storage space required for the solution as shown in Table 4-5. To review the other input used for the calculation, see Table 4-3 on page 58. In this calculation, we assume that there is a 10% increase in both the average e-mail size and the average number of e-mails per user per day. There is a direct correlation between these increases and the calculated storage estimated for each year.

Table 4-5 Hard disk requirements for Resource Manager and Library Server

Requirements:	**Year 1**	**Year 2**	**Year 3**	**Year 4**	**Year 5**
E-mails per day	300,000	330,000	363,000	399,300	439,230
E-mail per day after SIS	120,000	132,000	145,200	159,720	175,692
E-mails stored per year	31,200,000	34,320,000	37,752,000	41,527,200	45,679,920
Library Server database					
► DB2 space for CommonStore e-mails (GB)	74.10	155.60	245.26	269.78	296.76
Resource Manager database					
► Database size (GB)	10.17	21.36	33.66	37.03	40.73
► Object store (TB)	4.43	9.30	14.65	16.12	17.73
► NSE indexes on content (TB)[a]	0.66	1.39	2.20	2.42	2.66

[a] More than 1 TB requires usage of multiple item types.

E-mails per year: This is the total number of e-mails that have to be archived per year. We assume that there are five working days and 52 weeks in a year and it is based on the number of e-mails per day after SIS. See Figure 4-15.

```
E-mails per year
= E-mails per day after SIS * (days per week) * (weeks per year)
= 120,000 * 5 * 52
= 31,200,000
```

Figure 4-15 E-mail per year calculation

Library Server DB space for CommonStore e-mails: This is the disk space required to store the CommonStore e-mail records in the Content Manager Library Server database. When CommonStore archives an e-mail to the Content Manager repository, the following rows are created in the Content Manager Library Server database tables:

- One row in the root component table for each e-mail that is archived.
- *n* rows in the SIS child component table where *n* represents the number of recipients (plus senders) for each e-mail archived.

Let us assume that:

- Each row in the root component table takes 1800 bytes.
- Each row in the child component table takes 300 bytes.

To calculate the entire disk space required by the Library Server database, multiply the number of rows created in the root component table by the size of each row, multiply the number of rows in the SIS child components table by the size of each row and the average number of recipients per e-mail plus sender, and add these two numbers. See Figure 4-16.

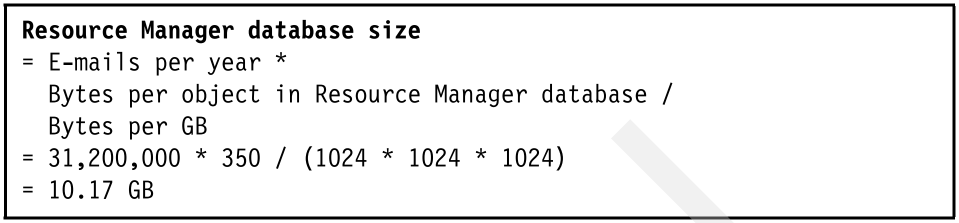
```
Library Server database space for CommonStore e-mails
= ((E-mails per year *
  Bytes per e-mail in root component table) +
  (E-mails per year *
  Average number of recipients per e-mail plus sender *
  Bytes per e-mail in child component table)) /
  Bytes per GB
= ((31,200,000 * 1,800) + (31,200,000 * 2.5 * 300)) /
  (1024 * 1024 * 1024)
= 74.10 GB
```

Figure 4-16 Library Server database space for CommonStore e-mails calculation

Resource Manager database size: This is the disk space required to store the CommonStore e-mail records in the Content Manager Resource Manager. When CommonStore archives an e-mail to the Content Manager repository:

- One row per e-mail is created in the Content Manager Resource Manager database table.
- Each row uses 350 bytes.

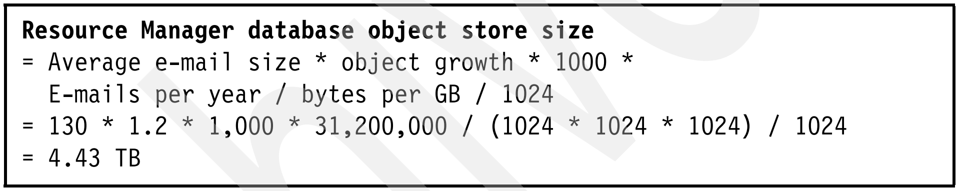
Figure 4-17 shows the calculation for the case study.



```
Resource Manager database size
= E-mails per year *
  Bytes per object in Resource Manager database /
  Bytes per GB
= 31,200,000 * 350 / (1024 * 1024 * 1024)
= 10.17 GB
```

Figure 4-17 Resource Manager database size calculation

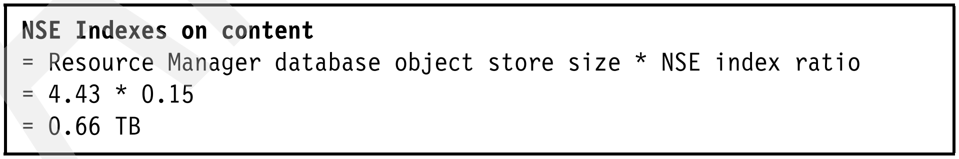
Resource Manager database object store size: This is the estimated disk space required to store the e-mail objects. It is calculated by multiplying the average e-mail size, the object growth factor, and the total number of e-mails per year. The object growth factor specifies the factor at which CommonStore converts rich text to plain text and store it in the object. Figure 4-18 shows the calculation for the case study.



```
Resource Manager database object store size
= Average e-mail size * object growth * 1000 *
  E-mails per year / bytes per GB / 1024
= 130 * 1.2 * 1,000 * 31,200,000 / (1024 * 1024 * 1024) / 1024
= 4.43 TB
```

Figure 4-18 Resource Manager database object store size

NSE indexes on content: This is the estimated disk space required to store full indexes of the e-mails. It is based on the Resource Manager object store size and multiplying that number by NSE index ratio. The NSE index ratio is the ratio of the full-text index to the original e-mail size. The smaller the e-mails, the higher the ratio. The range is from 6% to 10% based on some benchmark tests. Figure 4-19 shows the calculation for the case study.



```
NSE Indexes on content
= Resource Manager database object store size * NSE index ratio
= 4.43 * 0.15
= 0.66 TB
```

Figure 4-19 NSE indexes on content

For the next four years, we increase the total number of e-mails by 10%. This increases the corresponding disk storage requirement also by 10%.

Figure 4-20 shows the estimated disk space requirement for Library Server to store metadata and index information of e-mails in five years. The increase in the storage space is due to the estimated increase in the number of e-mails and the increase in the average e-mail size.

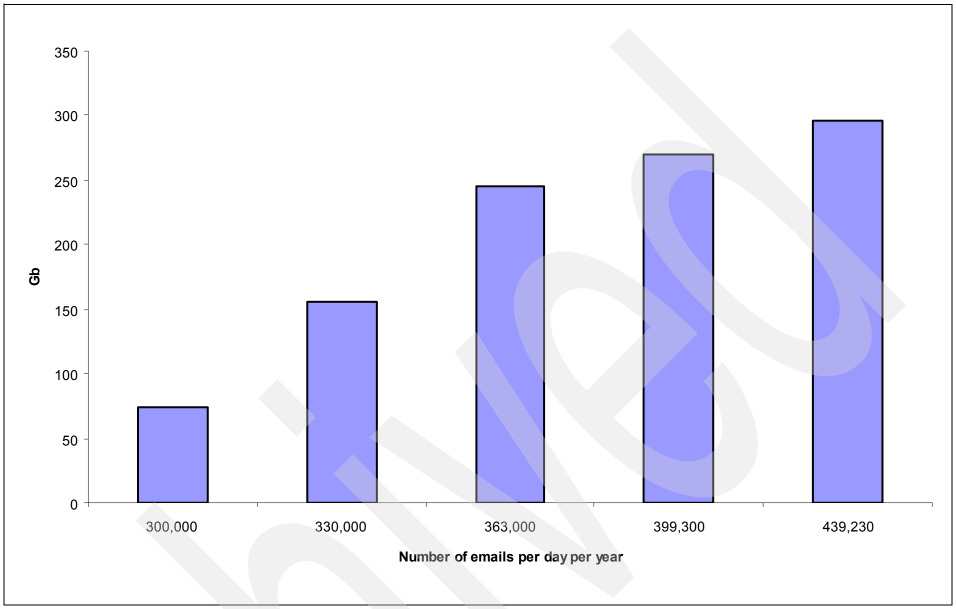


Figure 4-20 Library Server database requirements

Figure 4-21 shows the estimated disk space requirement for the Resource Manager database and object store over the five-year period.

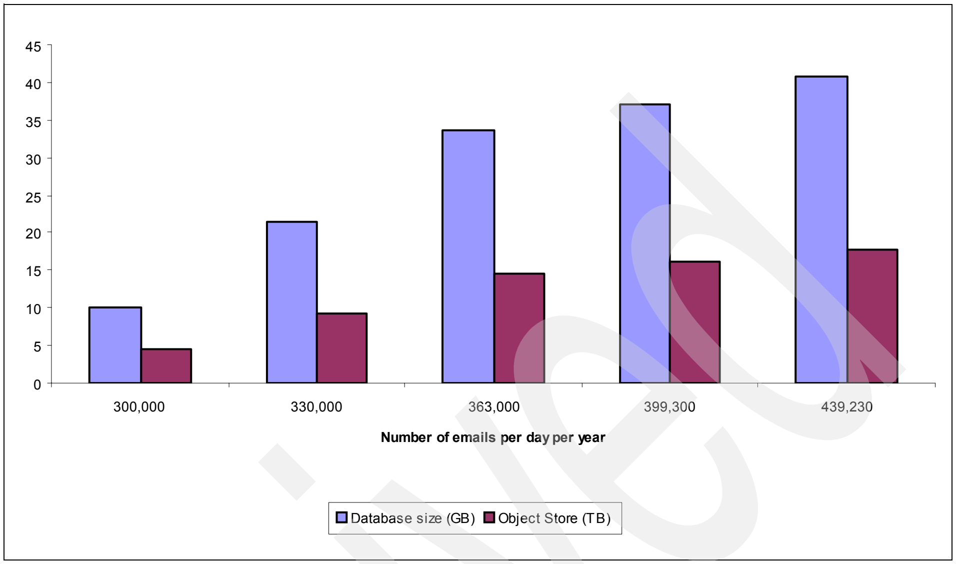


Figure 4-21 Resource Manager database requirements

The object store requirement represents the storage of the e-mail object files. The number is in TB and the Resource Manager database size is in GB.

In conclusion, sizing depends on the number of e-mails that you have to archive per day, the available hours of the CommonStore archiving operation, the average size of your e-mails, the attachment information (the number of the attachment, the average attachment size, the percentage of e-mails that have attachments), and the projected growth of the e-mails and their average size over years. Remember that the sizing tool is a living tool that the Content Management development and services team is constantly updating based on additional data and changed hardware. We recommend that you always obtain the latest sizing tool if you are eligible to get one, understand all the calculation behind it, and use it as the guidance to calculate your hardware requirements for the CommonStore solution. You can also use this tool to review your existing system setup to ensure that your CommonStore and Content Manager perform at optimal efficiency and prevent any problems that occur due to growth of your system. Consult with IBM specialists to finalize any system planning before you implement the solution.

5

Solution deployment and configuration

In this chapter, we discuss how to deploy and configure a CommonStore solution.

We do not cover eMail Search in this chapter. This topic is covered in Chapter 7, “eMail Search for CommonStore” on page 107. Configuration for CommonStore for Exchange Server (CSX) and Content Manager are also currently not covered in this version of the IBM Redbook.

We cover the following topics in this chapter:

- Overview of deployment and configuration
- CSLD deployment and configuration
  - Deployment of CommonStore for Lotus Domino servers
  - Configuration of CSLD and Domino server
- Hints for multi-threaded archiving
- Initial offload consideration
- Large system deployment consideration

5.1 Overview of deployment and configuration

The foundation of deploying and configuring a CommonStore solution is solution planning (see Chapter 2, “Solution planning” on page 17), solution design (see Chapter 3, “Solution design” on page 27), and solution sizing (see Chapter 4, “Sizing” on page 45). When you complete these three phases of the project, you can define the following items:

- A detailed solution architecture
- The sizing for the solution in terms of hardware sizing
- The hardware and software stack required to implement the solution

With this information ready, you can start to implement the solution by deploying the hardware and software components and by configuring the components.

The first part of the implementation is the deployment. The deployment of the CommonStore solution is the placement of the hardware (servers) in the appropriate locations based on the solution architecture. This server placement includes ensuring that the operating system is set up and configured properly with the latest updates and fix packs installed. Furthermore, we also include the installation of the software components on the dedicated servers into the deployment task.

The second part of the implementation is the configuration of the software components as outlined in the solution architecture.

In the remainder of this chapter, we assume that the network infrastructure is in place and available. If not stated otherwise, all the deployment and configuration is done on components connected within the same local area network (LAN).

5.2 CSLD deployment and configuration

When the Content Manager is deployed and configured, we deploy and configure CommonStore for Lotus Domino. In a larger project, it is most likely that there are different teams responsible for each of these tasks. However, the CommonStore for Lotus Domino deployment and configuration is a project phase that should not be shared among different teams. Within this phase, we perform the following tasks:

- Deployment of CommonStore for Lotus Domino servers
- Configuration of CommonStore for Lotus Domino
- Configuration of Domino server

Our objective is to share our real case experiences about CommonStore for Lotus Domino and Domino server. We provide in-depth information about running the different tasks of CommonStore, archiving policy setup, and testing. The impact on users especially during the initial offload phase is also covered in this chapter.

5.2.1 Deployment of CommonStore for Lotus Domino servers

In most of the CommonStore for Lotus Domino projects, there are dedicated hardware for the CommonStore for Lotus Domino servers. It is advisable for test and development systems only to use CommonStore for Lotus Domino on the Domino server or on the Content Manager server. The servers provide a Windows or an AIX platform for CommonStore for Lotus Domino.

Before you start the installation of the software, create a list of the software packages including the required updates and fix packs. We recommend that you use the latest software only. This is true not only for CommonStore for Lotus Domino itself, but also for the Content Manager client and for Lotus Notes and Domino server. With Lotus Notes and Domino, it is not the version but the Notes-API provided with that version that counts. CommonStore for Lotus Domino V8.3.1 uses Notes-API 3.0a (the latest version for CommonStore for Lotus Domino from September 2006 is using Notes API 3.0a HF1), which is part of Lotus Domino V6.0 and later versions. Always check the latest documentation for the latest version information.

Based on the software list, it is useful to create an installation medium such as a CD or a shared network drive. The CD or the shared network drive should also contain the license files, which are required during configuration. Another valuable document for the software installation is an installation checklist. This list not only contains the software packages to install but also the sequence of installation. If there are more than one CommonStore for Lotus Domino servers to install, follow this checklist and secure a similar setup of all servers. It can be helpful to save a default instance directory on each CommonStore for Lotus Domino server for further usage. The default instance directory is located for Windows in <*CSLD program path*>\server\instance01 (for example, C:\IBM\CSLD\server\instance01) or for AIX <*CSLD program path*>/ (for example, /usr/lpp/csld/bin/).

From our experience, it is not useful to invest a lot of time to create automated installation images. We recommend that you maintain a detailed documentation of the installation process and to have a recent installation medium available.

After the deployment of the CommonStore for Lotus Domino servers, configure these servers.

5.2.2 Configuration of CSLD and Domino server

The following list of steps guides you on how to configure CommonStore for Lotus Domino and Domino server.

For preparation:

1. Create a Notes user and ID file for the CommonStore for Lotus Domino user. In most cases, this is the account with access to all servers and all mail files. Because this Notes ID can access all mail file, make sure that it is not distributed freely in your organization, and use more than one, non-trivial password for this Notes ID.
2. Create user accounts for the CSLD Content Manager agents.
3. Update your mail templates with the design elements necessary for CommonStore for Lotus Domino.
4. Test your solution including the design of the enhanced mail template.

To configure the CommonStore for Lotus Domino server:

1. Create a folder in D:\IBM\CSLD called log.
2. Create a data1 folder in the instance01 subdirectory. This is the Notes data directory for the CommonStore for Lotus Domino tasks. Because CommonStore for Lotus Domino opens a Notes session in this folder, many of the typical Notes files are created. These files are not deleted automatically. Clear this folder from time to time.
3. Copy the CommonStore for Lotus Domino Notes ID and the notes.ini delivered with the CommonStore software package into the data folders and customize the ini file by following the guidelines of *CommonStore for Lotus Domino V8.3.1 Administrators and Programmer's Guide*, SH12-6742.
4. You require the same number of transfer directories as the number of database profiles that you are defining. The settings in the CommonStore for Lotus Domino configuration database documents must match these settings.
5. If you plan to install multiple CommonStore for Lotus Domino server instances, add extra directories required for the CommonStore for Lotus Domino server instances by copying the instance01 folder. Rename them to instance02 and instance03. Do not forget to rename the data folders inside these folders and to adopt the path settings inside the notes.ini files.
6. Create as many csservice.ini files as preferred (for example, server, task, and crawler specific) and give them descriptive names. Install the CommonStore for Lotus Domino Windows services.

7. Copy the archint_cm8.ini file from the instance01 directory into all the instance paths and rename it to archint.ini. Edit this configuration file and enter the specific values. Customize the csservice.ini and crawler.ini. Install CommonStore for Lotus Domino as a service.
8. Add the path of the nnotes.dll to the path variable in the system environment. Set the path for the CSNINSTANCEPATH environment variable to D:\IBM\CSLD\log. By doing this, all startup traces of the CommonStore for Lotus Domino tasks are separated from the traces of the task that you define in the database profiles. Mostly, only the latter helps you to solve problems.
9. Create the archint.cfg, csld.cfg, and the nodelock files. The archint.cfg contains the password (in encrypted format) for the Content Manager client accessing the back-end. Each instance directory contains an archint.cfg file. The csld.cfg contains the password (in encrypted format) of the Notes ID that is used for CommonStore for Lotus Domino. This file is created in the directory CSNINSTANCEPATH.

To configure the Domino server:

1. Create a group (for example, CSLD_ArchiveAdmins_Manager), which includes all exclusive administration users for the CommonStore for Lotus Domino related databases.
2. Customize the access control list (ACL) of the job database template and sign the template with a Notes signer ID valid in your Domino organization. Do the same with the configuration database template.
3. Create the required amount of job databases in the \csld subdirectory of the servers' data directory from the template.
4. Create a configuration database from the template in the \csld subdirectory of the servers' data directory. Optionally, create replicas of this database on all Domino servers belonging to the same location site and let them replicate using the hub servers.
5. Customize the design and the ACL for the mail database template and replace the design of all mail files in the Domino environment.
6. Perform the folder update of the mail database, if required.
7. Select **Do not mark modified documents as unread** on all mail files to avoid documents modified by CommonStore for Lotus Domino from being displayed as unread. In most of the cases, this can lead to some confusion for the users.

For the configuration database, do the following steps:

1. In the configuration database, create the following documents:
   - Database profiles
   - Document mappings
   - Special mappings (if necessary)
   - Content type mappings (if necessary)
2. If automatic archiving is used, create additional configuration documents:
   - Policies
   - Database sets
   - Scheduled tasks

5.3 Hints for multi-threaded archiving

When a message gets archived, it is processed by the following CSX or CSLD components:

- CSX or CSLD task, worker thread
- Archpro
- Content Manager agent
- (CSX or CSLD task, committer thread): Post-processing of message

Single versus multi-threaded archiving

For a single-threaded configuration (one worker, one archpro, one Content Manager agent), expect to see a little less than one mail per second archived (or 3,000 per hour).

For a better throughput, use a multi-threaded configuration. Keep the following information in mind:

- Adding more CSX or CSLD work threads do not help; you also have to add more Content Manager agents.
- Vice versa: Adding more Content Manager agents alone does not help; you have to add more CSX or CSLD work threads.
- Ten Content Manager agents do not give you 10 times the throughput of one agent: Expect that one archpro instance can give you up to four times the throughput of one Content Manager agent. Thus, configure the number of Content Manager agents in the range of 4 - 8 (and approximately the same number of CSX or CSLD task workers).

- One archpro instance gives you a throughput of approximately three messages per second (or 12,000 per hour).
- There is no data available showing that one archpro instance running on a multi-processor can provide a better throughput than the value mentioned previously.
- Tests have shown that a second archpro instance, configured with 4 - 8 Content Manager agents, running on the same 2-CPU machine can give a slightly smaller throughput. Thus, you get to 8 messages per second throughput.
- However, for a 4-CPU machine, you should not assume that the four archpro instances can give you four times the throughput of a single instance. Take a threefold throughput (12 messages per second or 36,000 messages per hour).

5.4 Initial offload consideration

When a CommonStore solution is introduced, it is almost never possible to start with the normal day-to-day business of archiving e-mails. In almost all deployments, you have to deal with a huge amount of historical data stored in the mail files. Hence, an initial offload is often the first phase for which the CommonStore solution is used right after its deployment. The initial offload is a process to archive the historical e-mail from mail files. As already stated, this normally happens during the introduction phase of a CommonStore solution or if the archiving policy changes dramatically resulting in larger amounts of e-mails to be archived. For instance, a CommonStore solution is enabled with a policy to archive all documents older than one year. That leads to an initial offload of e-mail older than one year, regardless when the database was created. If a mail file is a couple of years old with a size of 1 GB and 60% of the documents are older than one year, then there is an initial offload of approximately 600 MB from that mail file. In contrast to that, during the daily archiving, only a handful of e-mails are expected to be archived per mail file after the historical data is archived already.

From our experience, there are two approaches for the initial offload: A time-driven setup or a content-driven setup. When using a time-driven setup, you normally start to archive the oldest e-mails first, and then adjust the archiving policy in reasonable steps towards the final setting that will be used for the daily archiving. The content-driven approach requires you to have much more knowledge about the content of the mail files than the time-driven approach to create the right archiving policies. In case you know that there is a particular amount of a certain type of documents (for example, to be specified by Form name), these documents can be archived first, and then the archiving policy can be adjusted to match the next type of documents. The content-driven approach can be also based on special content such as subject fields always containing a special value or the size of the e-mails.

Regardless of the approach you choose, it is advisable to accompany the initial offload by attentive monitoring of the data throughput. The initial offload should under no circumstances influence the performance of the mail system to avoid dissatisfaction of the users.

5.5 Large system deployment consideration

Deploying an IT solution in a large, distributed enterprise environment is always a challenging task. A CommonStore solution is no different in such a situation. We distinguish between the two main areas of operation of a CommonStore solution: The archiving process and the retrieving/searching processes.

For the archiving process, answer the following questions before or during deployment:

- Where are the back-end components of the mail system located (that is, centralized or distributed)?
- Is the network capacity in place to support a distributed CommonStore solution?
- Is there a backup plan in place for the CommonStore solution? How will the recovery processes work in case of data loss?
- Can the CommonStore solution be monitored in a decent way?
- How are new mail systems integrated and can the CommonStore solution deployment deal with moving sources (for example, relocation of mail servers)?

For the retrieving/searching processes, the following questions are raised:

- Is the deployed solution capable of serving thousands of user requests when using document retrieval via HTTP?
- Does the solution address the needs of remote or offline users in terms of the support process?

- Can data be retrieved or searched even if a user is relocated to a department in another location?
- Will requests that are created offline be processed after reconnection to the network?
- How difficult would it be to re-create the original condition of a mail file with all e-mails if necessary?

In conclusion, deploying a CommonStore solution in an enterprise environment requires elevated concentration on automation, monitoring, and support for the solution.

6

Journaling

This chapter describes the journaling feature provided by Lotus Domino and Exchange Server that can be used within a CommonStore solution. Journaling becomes mandatory when there is a requirement to archive all e-mails passing through your company's e-mail servers.

We cover the following topics in this chapter:

- Journaling and its requirement
- System architecture
- Journaling process
- Journaling in CommonStore for Lotus Domino
- Journaling in CommonStore for Microsoft Exchange

6.1 Journaling and its requirement

The capture of every e-mail is vital to assist many organizations today in their legislative challenges. Some companies by law are required to capture all e-mails that are exchanged between employees and customers. For example, the Securities and Exchange Commission (SEC) requires businesses to retain all their messages related to the transactions that they undertake. In case of customer complaints, the internal information security department must provide the required information regarding the history of the customer service.

To capture every e-mail message that the mail server router processes and to store the e-mails in a separate database, enable the journaling feature provided by Lotus Domino and Microsoft Exchange.

When you implement an e-mail archiving solution for long-term e-mail retention and discovery, you can capture e-mails and keep them for a specific period of time. The solution must prevent the e-mails from being deleted by users unless the retention period expires. Journaling of users’ mailboxes provides a key to the solution. Additionally, when you implement this type of the solution, you should install eMail Search component and enable full-text indexing, to ensure that all e-mails are indexed, and that they can be searched and discovered when necessary.

6.2 System architecture

When journaling is enabled, the system makes a copy of each e-mail message as it passes through the user’s mailbox to its destination (or before it is received by users) and places the copy in the mail journaling database (for Lotus Domino) or journal recipient mailbox (for Microsoft Exchange). This copy of the e-mail message is retained. Even if the recipient, or an agent acting on the recipient's mail file, deletes the message immediately upon delivery, the copied e-mail remains in the mail journaling database or the journal recipient mailbox. Journaling is automatically performed by the server. Only authorized personnel can examine a journaling database (for Lotus Domino) or journal recipient mailbox (for Exchange).

Figure 6-1 shows a typical CommonStore solution architecture that uses journaling and eMail Search components. Both of these are optional, but should work together to provide a discovery-capable CommonStore solution.

Note that when journaling is enabled, CommonStore archives from the mail journaling database or journal recipient mailbox.

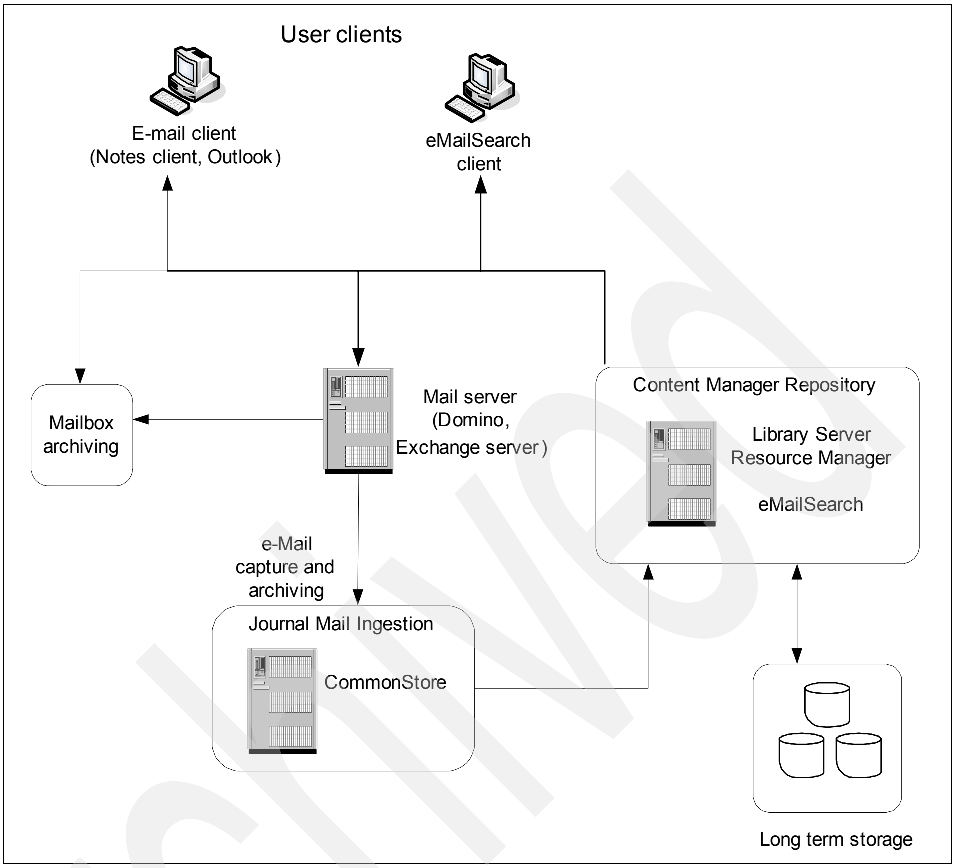


Figure 6-1 CommonStore solution architecture with journaling and eMail Search

There are four main components that make up the solution:

- Mail servers (either Domino or Exchange Server): The mail servers offer the following capabilities:
  - Mailbox management

    If the mail server performance decreases due to large volume of e-mails being stored on the mail server, you can set up quotas on the mailboxes. This means that when an e-mail database reaches this quota, the user of that mail database is forced to delete the e-mails manually or create a local archive database to move documents from the mail database. This reduces the size of the user's mailboxes, but it is completely the user's responsibility to maintain the local archive.

  - Mail journaling

    As mentioned previously, you might be required to capture all e-mails passing through your system. When journaling is enabled, the mail server automatically copies every e-mail into the journal database before it delivers the e-mail to the recipient.

- CommonStore server

  CommonStore automatically archives user mailbox or journaled databases into archived repository (in this IBM Redbook, we focus on Content Manager only). Note that stubbing documents in user mailboxes after archiving also reduces the size of the e-mail database on the server, thus reducing or even eliminating the need for local archiving.

  There is no requirement to have CommonStore components on the same server where journaling mail ingestion is present. Figure 6-1 shows a sample.

- The Content Manager repository consisting of Library Server and Resource Manager:
  - Library Server is a relational database. It stores e-mails attributes and full-text indexes of e-mails if full-text indexing is enabled.
  - Resource Manager is used to store and retrieve documents, images, and multimedia resources on file servers.
- User clients consisting of e-mail client and eMail Search client

  E-mail client provides users with access to their e-mails from their mail databases. It also provides local archiving functionality to move documents from mail databases on the server to local databases.

  eMail Search client enables authorized personnel to search among all archived and indexed e-mails.

6.3 Journaling process

For Lotus Domino, mail journaling works in conjunction with mail rules. Create a journaling rule to specify the criteria for selecting which e-mails to journal. For example, you can journal e-mails sent to or from specific people, groups, or domains. Before depositing messages in the mail journaling database, the e-mail router encrypts them to ensure that only authorized persons can examine them. Journaling does not disrupt the normal routing of an e-mail. After the router copies the e-mail to the mail journaling database, it dispatches the e-mail to its destination.

For Microsoft Exchange, mail journaling is enabled per message store. If you assign a journal recipient mailbox to a given message store, all mailboxes located on that message store are journaled. The way the messages are stored in the journal recipient mailbox depends on the journaling type: For message-only journaling, the journal recipient mailbox stores just a copy of the mail as it was sent to the recipients; for bcc journaling, all recipients are added to the bcc field of the message; for envelope journaling, the original message is wrapped within an envelope message that contains the delivery information. No encryption is done - authorization works by controlling access to the journal recipient mailbox.

Let us examine two different scenarios of journaling:

- Selective journaling in a single-mail server environment
- Journaling all e-mails in a distributed environment

Selective journaling in a single-mail server environment

In the first scenario, we discuss journaling on selective user mail databases in a single-mail server environment. Figure 6-2 shows the data flow.

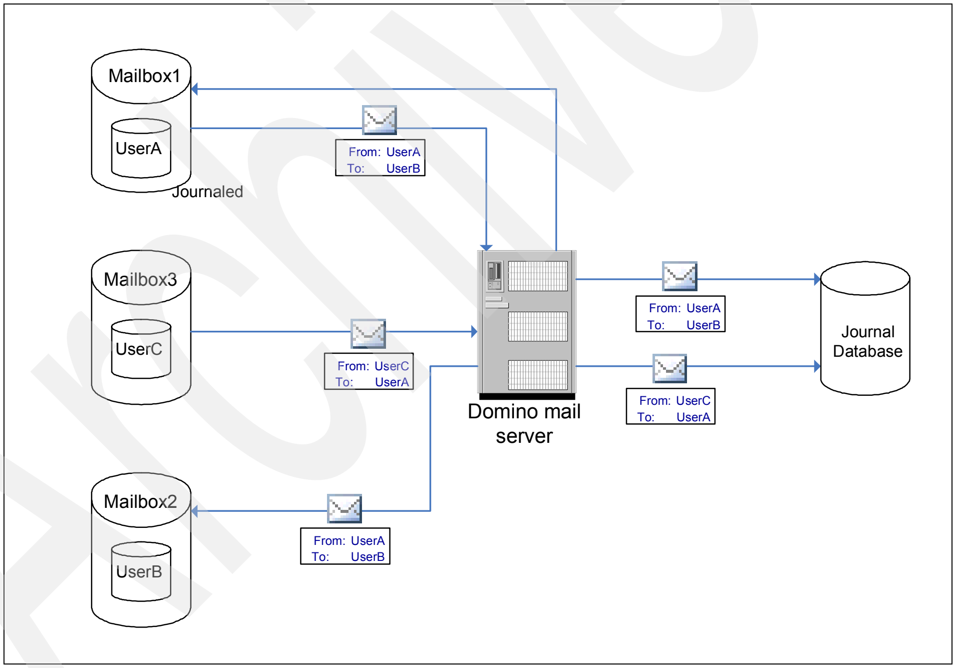


Figure 6-2 Journaling sample scenario

In this scenario, a company is only required to journal selective users, for example, userA. The user can be the CEO or the people who are in charge of the financial transactions and decisions. The mailboxes of other users, such as userB and userC, are not required to be journaled.

When UserA sends an e-mail to UserB, the Domino server detects this action and journals the e-mail from UserA. The server copies the e-mail from userA to the mail journaling database before delivering the e-mail to UserB.

When UserC sends an e-mail to UserA, Domino copies the e-mail into the mail journaling database before delivering the e-mail to UserA. Every e-mail that UserA sends or receives must first be copied to the mail journaling database before it is delivered to the appropriate destination.

When UserA deletes e-mails from userA's mailbox, the e-mails in the mail journaling database remain untouched.

Journaling all e-mails in a distributed environment

This scenario provides a more complex architecture of journaling on a distributed Domino server. Figure 6-3 shows the data flow diagram.

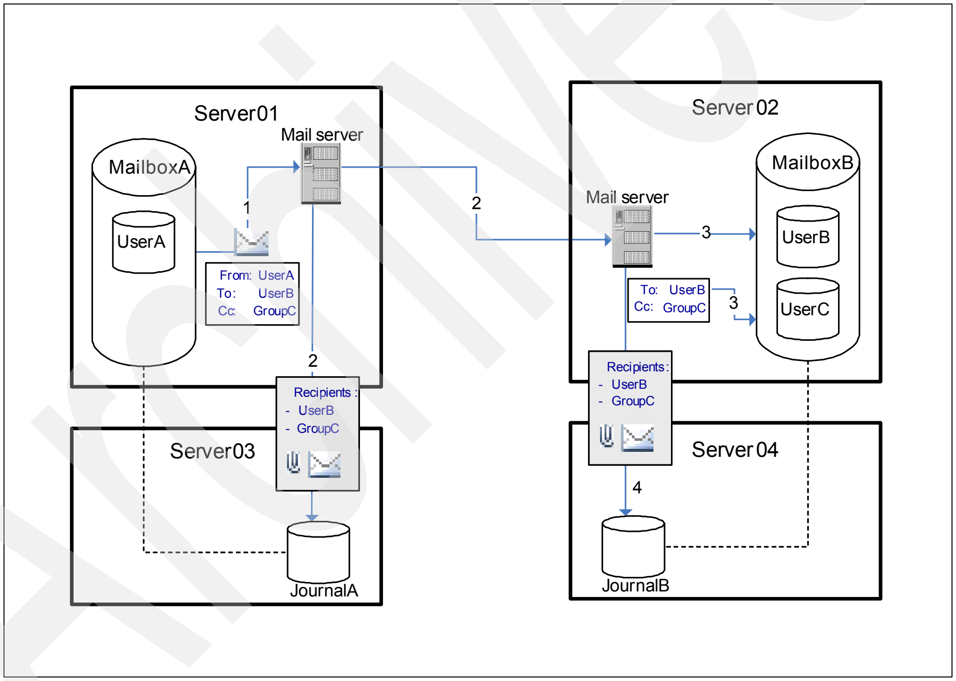


Figure 6-3 Journaling in a distributed Domino environment

In this scenario, journaling is enabled to capture all e-mail messages from all users in the company. There are multiple mail servers in the existing environment.

We show four servers in this environment:

- Server01 with mailbox of UserA
- Server03 with JournalA database to keep e-mails to and from UserA
- Server02 with mailboxes of UserB and UserC
- Server04 with JournalB database to keep e-mails to and from UserB and UserC

Although we show only one user mailbox per mail server, or one journaled mailbox per server, in reality, you always have multiple mail databases in one mail server and multiple mail journaling databases per server.

On Server01, in the e-mail server directory, GroupC exists, and it contains UserC.

When UserA sends an e-mail to UserB and GroupC, the e-mail goes to the mail server on Server01. This mail server knows that UserA is journaled in JournalA database. Before sending the e-mail to the mail server on Server02, the router (that routes e-mails) copies the e-mail onto JournalA database. It then sends the e-mail to Server02.

When the mail server on Server02 (in a different location with different domain) gets the e-mail, it stores a copy in JournalB on Server04 and delivers the e-mail to UserB. The mail server also checks in the e-mail server directory and obtains the receiver name in GroupC to send the e-mail to UserC.

The standard journaling feature captures e-mails delivered to UserB and UserC and stores them as one copy in JournalB.

For CSLD, journaling does not expand group names before copying e-mails. This means that the e-mail is journaled into the JournalB database with GroupC as the recipient without a UserC name. This can cause difficulties when searching specific e-mails delivered to the specified user, for example, UserC.

To solve this issue, you can implement eMail Search with a *group expansion* feature. Group expansion resolves the group names before copying them into a mail journaling database and storing them with additional attribute information in an archive.

For Microsoft Exchange, envelope journaling does some group expansion. CSX takes no additional action.

6.4 Journaling in CommonStore for Lotus Domino

There are two methods available for journaling e-mails in a Domino environment:

- Local journaling: Copying e-mails to a local mail journaling database
- Remote journaling: Forwarding e-mails to a mail-in database

In *local journaling* (also known as *copy-to-local database journaling*), the e-mail router copies e-mails to a mail journaling database on the same server where the user's mail database resides. Because local mail journaling database does not require e-mails to be transferred between originating or destination mail server, it minimizes the network traffic.

In *remote journaling* (also known as *mail-in database journaling*), you can journal all e-mails from multiple mail servers to one central location. Doing so might increase network traffic but makes the management of the mail journaling databases easier. Using this method also simplifies the CommonStore setup because you define less database sets for the crawler.

> **Important:** A word of caution, before you implement a solution with remote journaling, determine the network bandwidth and ensure that the throughput accommodates journaling of all e-mails.

Table 6-1 lists the differences between these two methods when implementing a CommonStore e-mail archiving solution.

Table 6-1 Mail-in and copy-to-local journaling methods

Methods	Remote journaling (mail-in)	Local journaling (copy-to-local)
Configuration	Allows one mail-in journal database for multiple Domino servers; simplifies the CommonStore setup, and you define less database sets for the crawler	CommonStore Crawler cannot resolve server groups, therefore, you must create as many database sets in the Configuration database for CSLD as the Domino servers.
BCC fields to store	BCC fields data is not captured	BCC fields data is captured
Encryption	Domino does not automatically encrypt e-mail messages. You have to use mail-in document with specified CommonStore user ID.	To encrypt e-mails, a fully qualified Notes name of the CSLD user is required.

Methods	Remote journaling (mail-in)	Local journaling (copy-to-local)
Single-Instance Storage (SIS)	Enabling SIS excludes duplicate e-mails; the system creates the same SIS hash for the e-mail sent by a person and received by another within the same domain.	At the time of writing, you require special fix packs to make SIS work with local journaling. Check with your IBM representative about the latest fix pack information.

6.4.1 Setting up journaling in Domino environment

Enable journaling from the configuration settings document in the Domino server. When enabling, specify where to store the journaled e-mails and set options for managing the security and size of the mail journaling databases.

After you enable journaling, the Domino server creates the mail journaling database.

Some of the information that you have to provide when enabling journaling include:

- Field encryption exclusion list: This specifies the e-mail fields that Domino does not encrypt when adding the e-mail to the mail journaling database. Encrypted fields cannot be displayed in a view. By default, the following fields are not encrypted: Form, From, Principal, and PostedDate.
- Journaling method:
  - Copy to local database is the default method. This is the local journaling mentioned previously.
  - Send to mail-in database; this is the remote journaling mentioned previously.
- Database name: This is the name of the mail journaling database. By default, it is MAILJRN.NSF.
- Encrypt on behalf of users: For CommonStore solution, use the CommonStore user IDs because CommonStore has to decrypt content for full-text indexing before archiving.

Note: In the CommonStore configuration database, in the *document mapping* section of the e-mail form, specify the Notes fields to be archived in the Content Manager as an attribute information for an e-mail. These fields must *not* be encrypted in the mail journaling database. We recommend that you check the configuration settings of the CommonStore e-mail form and to specify the same fields as in the journal configuration, and choose not to encrypt. For example, you should add the fields Subject and SendTo in the journal configuration settings.

It is also required that you select the rules of e-mail journaling. You can decide whether all e-mails of the users should be copied into the journaling databases or only those that meet specific criteria must be journaled.

6.4.2 Managing the size of the mail journaling databases

When you define journaling, configure how to manage the mail journaling database size:

- *Periodic rollover*: Rollover means to stop using the existing mail journal database and start using a new mail journal database for e-mails. Periodic rollover defines this rollover based on a specific period of days. The default period is one day. The mail journaling databases’ naming convention is MJ<*date*>.NSF, where <*date*> specifies the creation date in MMDDYYYY format of the database. Example: MJ09272006.NSF. If it is on a daily basis, there is one mail journaling database per day.
- *Size rollover*: This defines the rollover to be based on the mail journaling database size. When the existing mail journaling database reaches the specified size, a new mail journaling database is created and used. When you use this scheme, the naming convention for these databases is MJ<*XXXXXX*>.NSF, where <*XXXXXX*> represents a six-digit number, from 000001 to 999999 to indicate different databases.
- *Purge/compact*: This specifies when Domino deletes documents from the database and then compacts it.

Note: These methods are not applicable when dealing with remote journaling where you use mail-in database. In such cases, you must monitor the database size and ensure that archiving (using CommonStore) is taking place.

6.4.3 Group expansion feature from eMail Search component

If you use group names in address fields, it might be challenging to search for e-mails in archives using the user name that belongs to the group, because standard Domino stores e-mail fields as it is without resolving group names. It is possible to solve this issue using the group expansion feature that comes with eMail Search component.

The group expansion feature comes with an agent that processes the e-mails before they are journaled and expands any group names into the recipient names. The group expansion agent saves this information in the e-mail by creating the new fields in the document, but the To, CC, and BCC fields remain unchanged. When you configure the CommonStore index fields on the Library Server, you can include these new fields for searching.

When you implement a mailbox management together with journaling and group expansion, it doubles the size of the storage required for e-mails, because CommonStore always stores two copies of the same e-mail (if the e-mail contains group names) in the archive. This is because the hash that is created by the group expansion agent to resolve group names is different from the hash of the e-mail in the mailbox. The group expansion feature works only for domino server group names, not for local user groups.

The group expansion is shipped in one Lotus Notes template database. The database agent is developed and tested on Lotus Domino Version 6.5.4.

6.4.4 Performance considerations

Journaling-related deployment variables and the system configuration can impact the performance of the overall CommonStore solution.

Keep the following list in consideration when you set up journaling in your solution:

- If you have to use content-driven rules to determine what e-mails to journal, limit these rules in text and especially in text in body fields, because they can slow down the speed of journaling.
- Journaling rules must be simple.
- Besides the journaling rules that determine what e-mails to journal, the biggest deployment decision that affects the amount of e-mails that are journaled is what servers to enable journaling. This depends on your business requirements. If your company only wants to keep copies of e-mails from external sources or check whether the employees are e-mailing out proprietary information, enable journaling on the edge servers that connect the company to the Internet. If your company wants to capture all of the employees' e-mails, you have to enable journaling on all the mail servers. If you need to journal multiple groups of employees and the number of people is large, it might be better to move their databases to one server and keep journaling in one place. If it is a small number of people, you can place a journal setup for an individual person.
- Delivery of e-mails to a remote journaling server increases messaging traffic. Every e-mail goes to one extra recipient when journaling on a separate server. One of the benefits of remote journaling is that if the remote server is in a locked room, there is less of a security risk if someone tries to crack open the mail journaling database.
- An alternative to remote journaling is to move that data using alternate mechanisms such as replication and copy. This makes more sense if the later tasks such as e-mail archiving are removed from the production mail servers. To reduce load in this scenario, use local journaling and replicate the mail

journaling database to a central location based on a schedule or do it overnight rather than immediately. Rollover options are a useful conjunction to this. For example, periodic rollover forces the creation and usage of a new mail journaling database every day. You can safely replicate the non-active mail journaling database to a central location for further e-mail archiving.

- Third-party products that query or move data from the journal database might add load to the journal server as an add-in task. This can impact the mail servers depending on the configuration that is chosen. Running the integration on a remote server helps to lessen the impact.
- There might also be unexpected impacts from other software. For example, depending on the priority of rules, anti-virus or anti-spam software might intercept the e-mail before it gets journaled or the journal database itself might be scanned and the files removed. Configure the environment with this in mind.

6.4.5 Sample journaling solution

In this section, we present a sample CommonStore solution that uses journaling.

Primary requirement

A company wants to implement a CommonStore solution. They are required to keep all e-mails for seven years.

Current environment and specific requirements

The company has 3,000 users and is currently running 10 clustered Domino mail servers in a distributed environment. On average, there are 40 e-mails received or sent per user per day. Sixteen operation hours are available for e-mail archiving.

Solution

Each mail server is configured to have its own mail-in journal database to capture e-mails from all users in the server. A CommonStore for Lotus Domino server is installed on the 4-way Windows machine and configured to run two crawlers:

- One for journal tasks to archive all users' journal databases
- One for mail database tasks to apply mailbox management archiving

Content Manager is used as an archive repository for the e-mails. It is installed on a 4-way Sun™ Solaris 9 machine with 8 GB RAM. For long-term storage, Tivoli Storage Manager is used in conjunction with Content Manager.

This solution supports archiving of 120,000 e-mails per day. Figure 6-4 shows the architecture of the implementation.

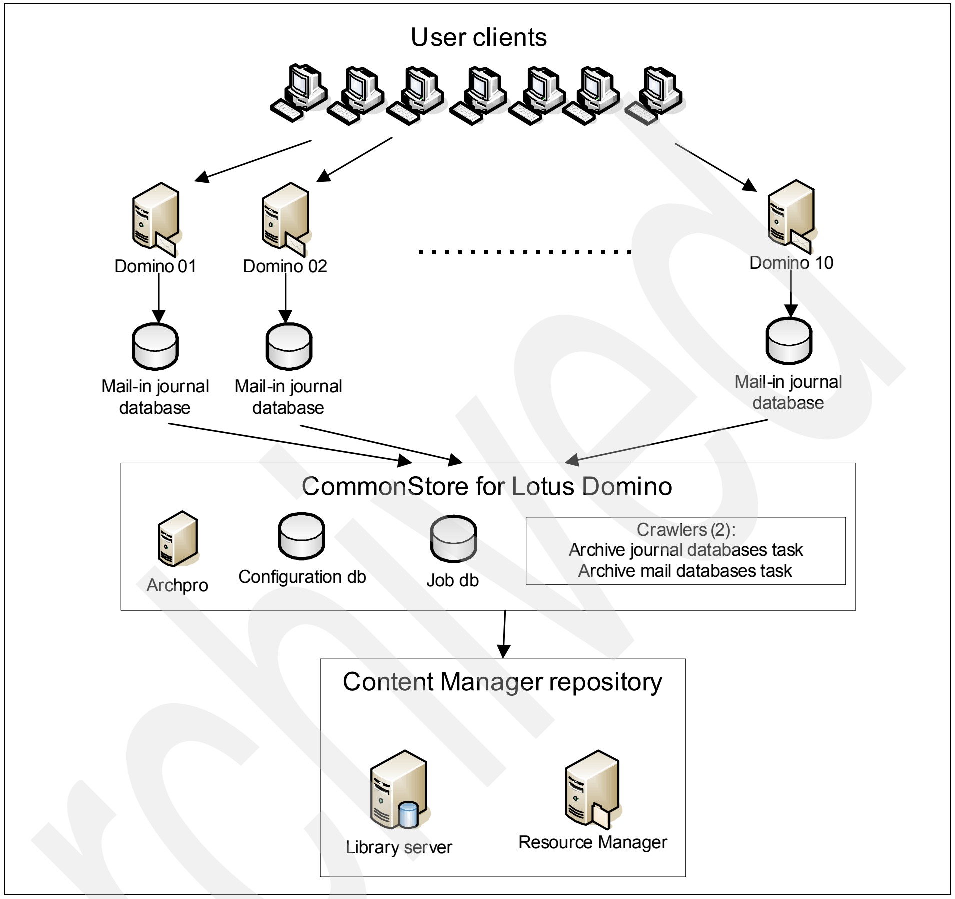


Figure 6-4 CommonStore solution for compliance

6.5 Journaling in CommonStore for Microsoft Exchange

There are three types of journaling with Microsoft Exchange server:

- *Message-only journaling* creates a copy of the original e-mail message in the journal recipient mailbox. The message header contains only the message recipient data that the sender declared to the recipients. Messages that are received from the Internet contains a message header. This header information is used by the message transfer agent to deliver the mail. This copy does not contain the recipients from the BCC field, the recipients listed in forwarding rules, or the recipients resulting from the expansion of the user groups (distribution lists). You can enable message-only journaling for any Exchange Server version.
- *BCC journaling* creates a copy of the original e-mail in the journal recipient mailbox. This copy contains the recipient addresses from the BCC field, but it does not contain recipients from forwarding rules or groups. You can enable BCC journaling for an individual server by adding a registry key and restarting the Simple Mail Transfer Protocol (SMTP) service and the Exchange Information Store service. The BCC field always contains the complete list of recipients (including the recipients from the To and CC fields). You can enable BCC journaling for Exchange 2000 and Exchange 2003 servers if the recipient list includes hidden distribution lists, query-based distribution list, or distribution lists that are expanded on another server. The recipients for these lists are not included in the journalized mail.
- *Envelope journaling* creates an envelope message in the journal recipient mailbox. The body of this envelope message (called journal report) contains a recipient list including the recipients from the To, CC, and BCC fields. Distribution lists that are internal to the Exchange organization are enumerated. In an envelope journaling message, the original message is included as an attachment. To enable or disable envelope journaling for the entire Exchange organization, you can run the exejcfg.exe program on one Exchange server. Envelope journaling requires the Exchange Server 2000 Service Pack 3 and the Exchange 2000 envelope journaling software update or the Exchange Server 2003 Service Pack 1.

> **Note:** To enable envelope journaling on the Exchange organization, run the **`exejcfg –e`** command. For further information, see the appropriate Microsoft Exchange documentation.

With CSX v8.3.0, envelope journaling was not supported. CSX v8.3.1 has two different policy types, one is dedicated for envelope journaling.

With CSX v8.3.1, if you use Exchange envelope journaling and you associate a standard policy to a journal recipient mailbox, the messages in that mailbox will not be archived and the crawler log will show that. If you associate an envelope journaling policy to a normal mailbox or to a journal recipient mailbox without enabling Exchange envelope journaling, the messages in that mailbox will not be archived and the crawler log will show that. If you associate an envelope journaling policy to a journal recipient mailbox with Exchange envelope journaling enabled, everything will work fine including indexing, text search and SIS.

As a conclusion: if you want to archive envelope journaled messages, you must use CSX 8.3.1 for best functionality.

You can use Content Manager or OnDemand as the back-end archived repository. To implement a CommonStore solution for long term e-mail retention and discovery, use Content Manager only.

You must define a special set of attributes or database fields for the archived repository that you use. Content Manager requires you to configure a child component and add the additional attributes to the child component. The CommonStore for Exchange Server handles these attributes internally so that mapping of journal message properties to archive attributes is not required. This means that you do not have to add property mappings to the archive object relating to the archived repository.

6.5.1 Setting up envelope journaling in Exchange environment

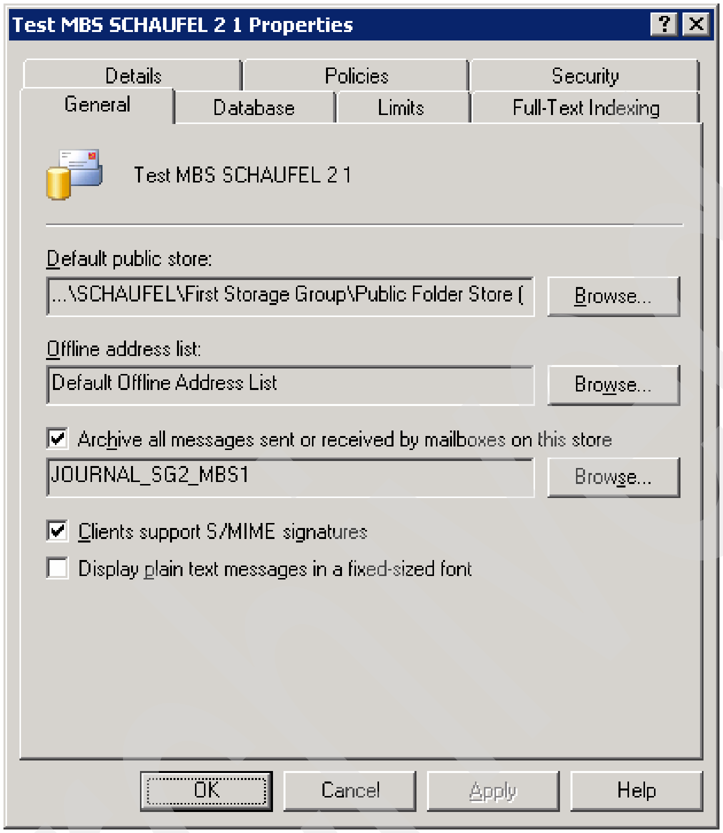
To enable journaling, perform the following steps:

1. Create an Exchange MailboxNote. This mailbox is a standard user mailbox and it does not have any specific settings.
2. On the Exchange System Manager, select the Mailbox Store to be journaled.
3. Right-click and select **Properties**.
4. Select the **General** tab.
5. Select the **Archive all messages sent or received by mailboxes on this store** check box. Note that it reads Archive, but in fact it means Journal.
6. Select the journal recipient mailbox.

   > **Note:** This does not control the journaling type.

7. Enable envelope journaling as described previously.

Figure 6-5 shows the creation of an Exchange mailbox with journaling enabled.



Test MBS SCHAUFEL 2 1

Default public store: ...\SCHAUFEL\First Storage Group\Public Folder Store (

Offline address list: Default Offline Address List

☑ Archive all messages sent or received by mailboxes on this store

JOURNAL_SG2_MBS1

☑ Clients support S/MIME signatures

□ Display plain text messages in a fixed-sized font

Figure 6-5 Exchange mailbox with enabled journaling

Figure 6-6 shows that UserA has a mailboxA on Server01, where journaling is enabled. UserA sends a message to UserB. UserB's mailbox is on Server02 in the mailboxB where journaling is not enabled. Server03 is a mailbox server that hosts only one journaling mailbox called JournalA. Server03 is a dedicated journal recipient mailbox server. Exchange performs several actions before a message leaves the sender's Exchange server. Exchange sets a journaling property on the message that identifies it as a journalized message. This property travels with the message to its various destinations in the Exchange organization.

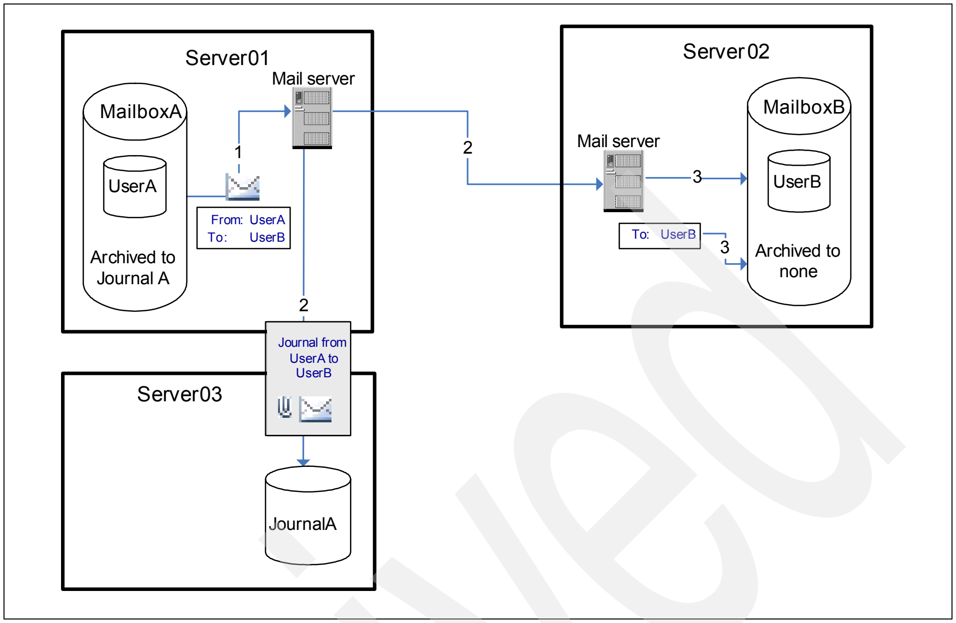


Figure 6-6 Journaling for a single mailbox database

Important: For further information about journaling, see the following Web site:

`http://www.microsoft.com/technet/prodtechnol/exchange/guides/E2k3Journal/`

CSX System Manager

The CSX System Manager is a Microsoft Management Console (MMC) that provides a graphical user interface for the CommonStore administrator. Using the CSX System Manager, the administrator configures the instances of the CSX task.

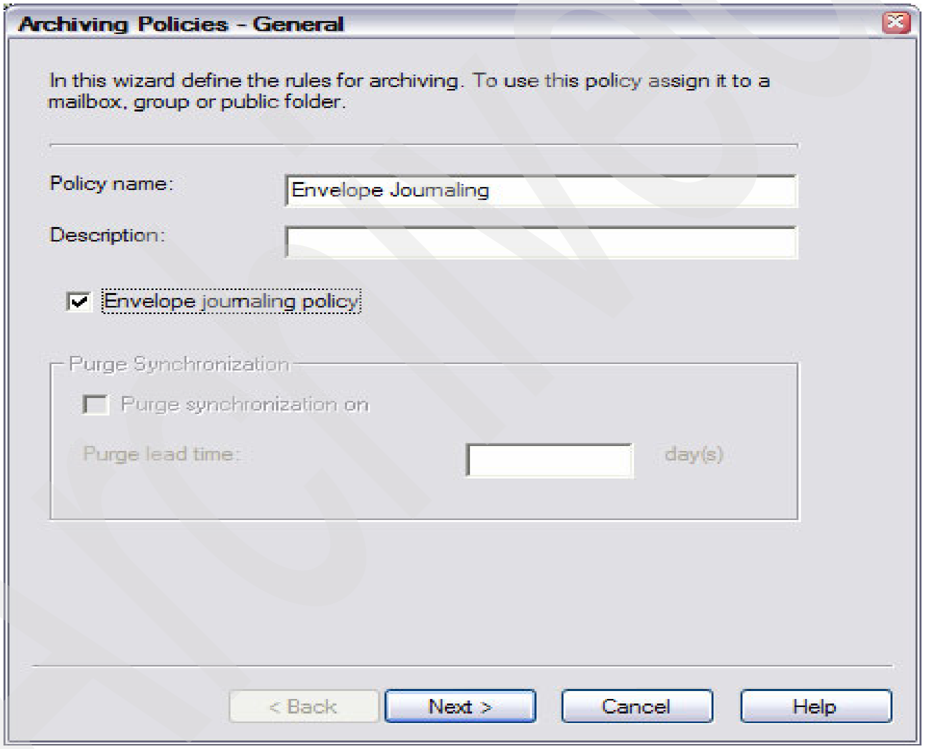
This includes the following activities:

- Specifying the number of threads to run, to control the degree of parallelism
- Specifying the list of Exchange 2003 servers to work on
- Specifying the job folders to poll
- Defining and maintaining policies for automatic archiving
- Defining and maintaining mappings

The CSX System Manager saves the settings in the Active Directory of the domain to which the CSX task instances and the connected Exchange 2003 servers belong to. When an instance of the CSX task is started, it reads the configuration data from this Active Directory. It is advisable to install the CSX System Manager on the workstation of the CommonStore administrator. This workstation must have access to the Active Directory of the Windows 2003 domain hosting your Exchange 2003 organization, and therefore must be a part of this domain.

When you configure Exchange to do *envelope journaling*, you need to associate an envelope journaling policy to the journaling mailbox.

Figure 6-7 shows the General properties of an archiving policy using the CSX System Manager. To create a policy to archive from an envelope journaling mailbox, select the **Envelope journaling policy** check box.



Archiving Policies - General

In this wizard define the rules for archiving. To use this policy assign it to a mailbox, group or public folder.

Policy name: Envelope Journaling

Description:

☑ Envelope journaling policy

Purge Synchronization

□ Purge synchronization on

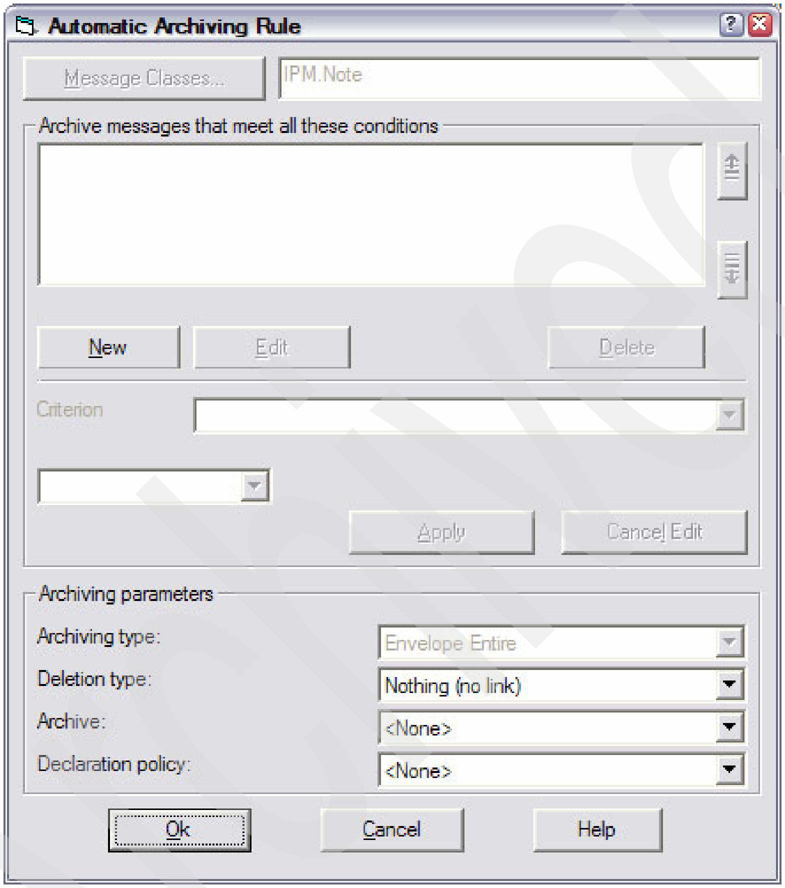
Purge lead time: day(s)

Figure 6-7 CSX System Manager: Policy object (1/2)

Envelope journaling policies only archive envelope journal messages; all other messages will be ignored.

As journaling mailboxes are not supposed to be used by users, client functionality is not available for envelope journaling policies. Therefore, settings related to interactive archiving, purge synchronization, and stubbing are disabled in envelope journaling policies.

Figure 6-8 shows the Automatic Archiving Rule page of an envelope journaling policy using the CSX System Manager.


Automatic Archiving Rule

Message Classes...: IPM.Note

Archive messages that meet all these conditions

Criterion

Archiving parameters

Archiving type: Envelope Entire

Deletion type: Nothing (no link)

Archive: <None>

Declaration policy: <None>

Figure 6-8 CSX System Manager: Policy object (2/2)

Note that some of the fields are predefined and cannot be changed:

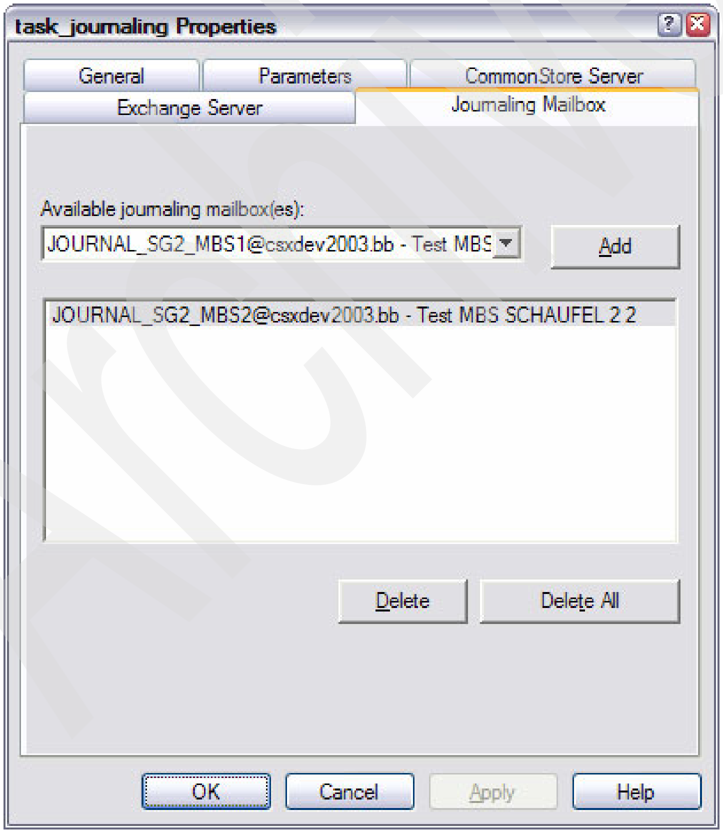
- As all envelope messages have the message class *IPM.Note*, no other message class can be selected.
- The archive type is always *Envelope Entire*.
- The deletion type is restricted to *Message* to remove the message after successful archiving and Nothing (no link), which only adds archiving properties to the message.

As journaling mailboxes typically process a high volume of messages, these mailboxes are handled individually, independent of the journaling method. While you associate all mailboxes and public folders of one Exchange Server to a CSX Task, you can assign the journaling mailboxes individually. This allows you to spread the workload on different workstations, if required.

When the CSX Task is started, it launches several threads to perform crawling. While all mailboxes for a given server are handled by one thread, and all public folders for that server are handled by a second thread, one additional thread is started for each journaling mailbox associated with the CSX Task. This ensures that journaling mailboxes are crawled continuously.

For example, if you assign two Exchange Servers and three journaling mailboxes to a task object, starting the Task will launch seven crawler threads.

Figure 6-9 shows the Journaling mailbox page of a task object using the CSX System Manager.



Available journaling mailbox(es): JOURNAL_SG2_MBS1@csxdev2003.bb - Test MBS

JOURNAL_SG2_MBS2@csxdev2003.bb - Test MBS SCHAUFEL 2 2

Figure 6-9 CSX System Manager: Task object

6.5.2 Performance considerations

In the receiving case, extra processing (beyond reading the journaling property) is required only when the receiving server is the expansion server for the distribution list or when the distribution list is hidden or query-based.

You can estimate the effect of journaling on a mailbox database by assuming that the enabled mailbox database can process approximately half of the messages that are sent, if all other conditions, such as CPU power, bandwidth, storage space, and disk speed remain constant.

Some factors that affect the storage requirements and performance characteristics of the journal recipient mailbox include:

- Envelope message overhead

  There is a small overhead for each journalized message, because the envelope journal message includes a plain text report with the original message attached. Unless there are thousands of recipients listed for a particular message, the plain text report adds less than 1 KB of overhead, in addition to the original message.

- Multiple journalized messages

  If expansion servers are used for a distribution list or alternative recipients are added as a message flows through your system, there are multiple instances of the same journal message, each reporting different recipients. Calculating the overhead that this produces is difficult because it depends on whether your organization uses expansion servers or hidden distribution lists, and this also depends on the habits that users have for sending their mail.

- Storage group configuration

  Limit the number of storage groups per mailbox database.

7

eMail Search for CommonStore

This chapter describes IBM eMail Search for CommonStore used for long-term e-mail retention and discovery purposes.

We cover the following topics in this chapter:

- eMail Search and its requirement
- eMail Search capabilities and user interface
- eMail Search system architecture
- eMail Search installation and configuration

7.1 eMail Search and its requirement

IBM eMail Search (eMS) for CommonStore provides the discovery feature that helps organizations to respond to litigation or regulatory agency requests. Using eMail Search along with journaling, organizations can capture and archive all the incoming and outgoing e-mails before they reach the recipients. After e-mails are captured, they are stored, full-text indexed, and are available to be searched by authorized personnel. In addition, eMail Search helps to retain e-mails for a specified period of time and deletes them after the retention period expires.

E-mails are not only communications between employees and with external people, they also contain business discussions, decisions, and important business documentations. E-mails represent corporate information asset. Depending on the type of the business, organizations might be required to keep e-mails for five years, seven years, or for shorter or longer periods of time. E-mails are subject to discovery and audit for legal or regulatory purposes. Often, organizations are not sure what information is going to be requested or when. Requests are typically part of a reactive process. There is usually a rush to find the information. When the information is found, it is preserved and reviewed by individuals, typically senior management, human resource personnel, and the legal team. After reviewing the information, it can be exported and passed to the appropriate individual.

The requirements for electronic discovery solutions are driven by organizations having to ensure compliance with rapidly growing and evolving legal regulations and standards for governance worldwide.

When evaluating an electronic discovery solution, organization requirements are split into two areas, e-mail storage and management, and search and discovery. The common themes for these two requirements are described in the following section.

E-mail storage and management requirements:

- Ensure that e-mails are captured and archived correctly and consistently.
- Ensure that e-mails are kept and not altered for a required period.
- E-mails are assets when they are kept and managed correctly.
- E-mails become liability when they are not managed properly, for example, they are either deleted too early or too late.
- Content can be effectively searched, reviewed, screened, and exported to support discovery requests.

E-mail search and discovery requirements also include:

- Fast and reliable access to archived e-mails:
  - Organizations usually do not know what they have to search for. This means that we have to provide full-text indexing on e-mails to provide search capability. In addition, the solution must provide flexible queries so that one can perform a search based on legal or business requirements.
  - Organizations usually do not know how often they have to conduct search for discovery cases. This means that the search engine for the solution has to be scalable.
  - If the organizations have to keep large volume of e-mails, the solution must provide scalable e-mail repository.
  - Many organizations want to have the solution in place just in case they are facing some legal challenges and have to respond to discovery orders.
- Typically small number of users of the search system with large result sets
- Queries are usually text-oriented, involving names and date ranges
- Simple to use intuitive interface
- Full audit trail
- Ability to export results into industry standard formats

CommonStore solutions with the eMail Search feature help organizations to meet the requirements detailed previously. Using the eMail Search feature, organizations can produce a discovery search across all e-mails that are sent or received using granular search methods.

Usually, a litigation request is precise and an organization is required to produce results in a short time. An example of a litigation request can be:

```
Provide all emails which have been sent or received by ‘Jon Doe’ to all
staff from the organization ‘ITSO’, which include the phrase ‘Shares’
either in the body or the attachment.
```

eMail Search enables the administrators with the right authority to perform this search across the archive, and export the results into multiple formats to be passed back to the requesting party. An audit trail can also be produced based on all of the user activities.

eMail Search for CommonStore also provide an option to save common searches. Search results can be saved in folders. The results can be stored and viewed multiple times.

7.2 eMail Search capabilities and user interface

eMail Search offers the following capabilities:

- *High-speed document indexing*: The application enables full-text indexing of e-mails at high speed, beyond the usual speed that the out-of-box NSE provides.
- *Domino group expansion*: This enables expanding of the group e-mail addresses internally when e-mails are sent or received using group address. The group e-mail address is internally expanded such that the archived e-mails have the associated individual user e-mail addresses. This enables the system to search for e-mails to and from a particular user even for the e-mails that the user received as a group.
- *Domino mailbox cleanup*: This is the EMail Purge component. It is used to delete archiving stubs from Lotus Notes. For more information, see section 4.5 of eMail Search documentation.
- *Domino e-mails expiration based on addresses*: Although still in development at the time of writing, this feature is targeted to be available in the very near future. It allows you to calculate the expiration date for Domino e-mails based on addresses. An XML file is prepared by the customer which contains the retention period for each address. During group expansion, this file is used to create a new document property, which can then be stored in Content Manager as a new attribute. Based on this new attribute, the e-mails will be expired accordingly. Check with the IBM representatives for the latest development on this new feature.
- *Outboard conversion server*: When running the Library Server (and INSO filters) on AIX, if it is necessary to convert either MS Binder files (.obd) or WinRAR files (.rar), we send the object to a Windows server, convert it, and return the text to the CommonStore exit for inclusion in the XML document that is indexed.

User interface

You can launch the eMail Search client using the following URL:

`http://<host name>/eMailSearch`

Figure 7-1 shows the initial logging window.

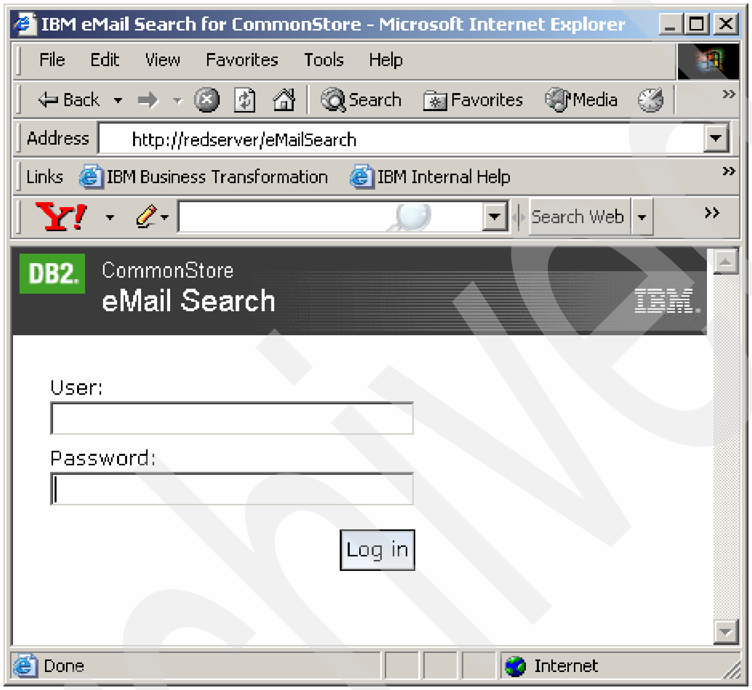
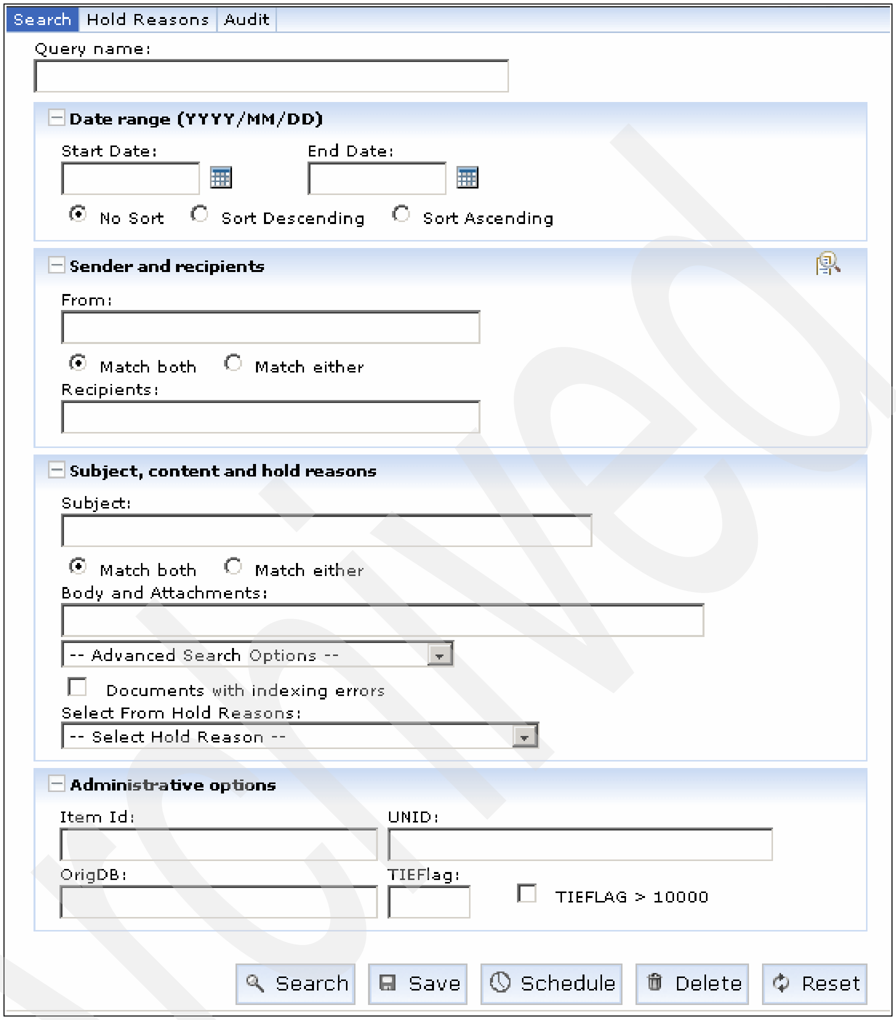

IBM eMail Search for CommonStore - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://redserver/eMailSearch
Links IBM Business Transformation IBM Internal Help
Search Web
DB2.
CommonStore
eMail Search
IBM
User:
Password:
Log in
Done
Internet

Figure 7-1 eMail Search login window

After you log in, you are presented with a search window. See Figure 7-2.



Query name:

Date range (YYYY/MM/DD)

Start Date:

End Date:

● No Sort ○ Sort Descending ○ Sort Ascending

Sender and recipients

From:

● Match both ○ Match either

Recipients:

Subject, content and hold reasons

Subject:

● Match both ○ Match either

Body and Attachments:

-- Advanced Search Options --

□ Documents with indexing errors

Select From Hold Reasons:

-- Select Hold Reason --

Administrative options

Item Id:

UNID:

OrigDB:

TIEFlag:

□ TIEFLAG > 10000

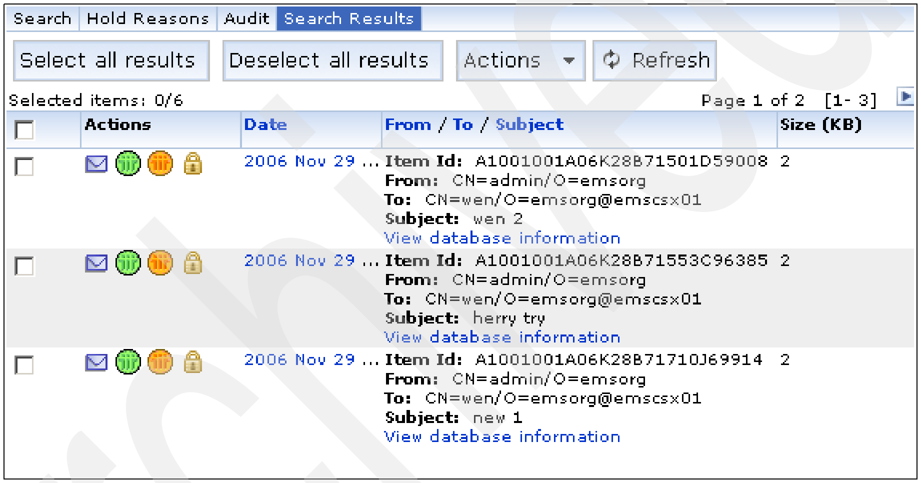
Figure 7-2 Search window for eMail Search

Notice, in this window, you can specify the start date and end date of the e-mails that you want to search. You can enter the sender and recipient information, the subject to be searched for, and other options.

At the bottom of the window, you can execute the following actions by clicking one of the buttons:

- *Search*: Execute the search action based on the input that you entered in Figure 7-2.
- *Save*: Save the input for the query for later search action.
- *Schedule*: Schedule the search/export/hold task periodically, for example, over the weekend, or in the evening.
- *Delete*: Delete the saved search query.
- *Reset*: Reset the window.

Figure 7-3 shows a sample search result window.



Selected items: 0/6

Page 1 of 2 [1- 3]

□	Actions	Date	From / To / Subject	Size (KB)
□		2006 Nov 29 ...	**Item Id:** A1001001A06K28B71501D59008 **From:** CN=admin/O=emsorg **To:** CN=wen/O=emsorg@emscsx01 **Subject:** wen 2 View database information	2
□		2006 Nov 29 ...	**Item Id:** A1001001A06K28B71553C96385 **From:** CN=admin/O=emsorg **To:** CN=wen/O=emsorg@emscsx01 **Subject:** herry try View database information	2
□		2006 Nov 29 ...	**Item Id:** A1001001A06K28B71710J69914 **From:** CN=admin/O=emsorg **To:** CN=wen/O=emsorg@emscsx01 **Subject:** new 1 View database information	2

Figure 7-3 Search result window for eMail Search

Content Manager hold

Using features of Content Manager, eMail Search can hold documents to prevent deletion even when the expiration date of the e-mails arrives. E-mails can be held for any number of reasons. Searching can be limited to a selected hold reason.

In Figure 7-3, we show three result entries that have hold on them. In this case, you can release the hold.

7.3 eMail Search system architecture

eMail Search for CommonStore runs under WebSphere Application Server and works with Content Manager. It is built on a number of components.

From a high level, these components are:

- Content Manager V8.3 with fix pack 3 plus test fix 1 of Content Manager V8.3 fix pack 3 or later
- CommonStore for Lotus Domino V8.3.1 or CommonStore for Exchange Server V8.3.1
- eMail Search server and client

Figure 7-4 shows the high-level architecture of eMail Search.

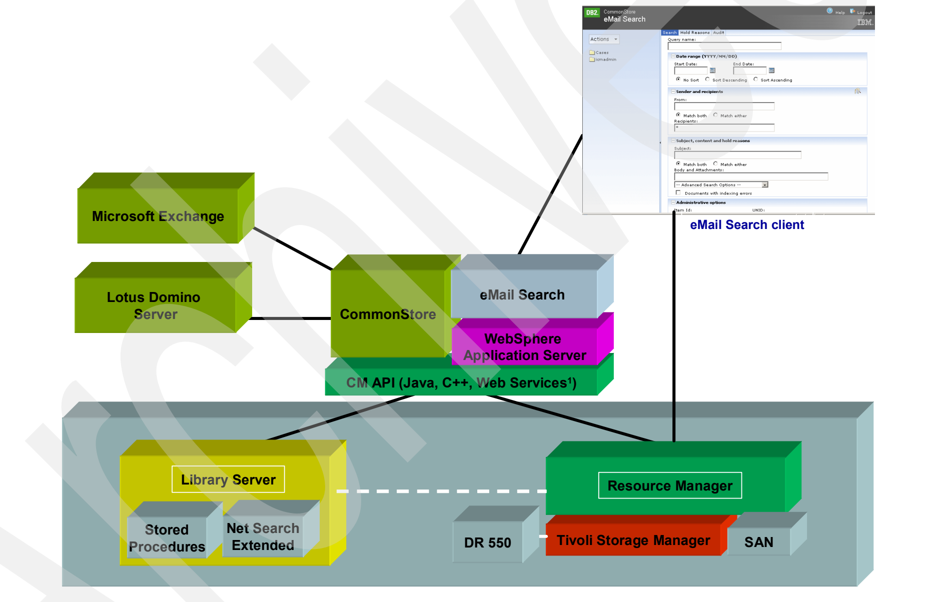


Figure 7-4 eMail Search high-level architecture

Content Manager

The Content Manager is a secure, scalable content management repository, which is used to store a wide variety of digital content. As part of eMail Search, the Content Manager stores all e-mails that are passed over from CommonStore.

Content Manager consists of two distinct parts:

- Library Server
- Resource Manager

The Library Server contains the index information of all e-mails added into Content Manager. Index information is both in the form of metadata, for example, subject and date and a full-text index. The Library Server (used with eMail Search) supports IBM DB2.

The Resource Manager stores the e-mail objects added into Content Manager. When objects are added into Content Manager using CommonStore, Single-Instance Storage (SIS) can be applied. SIS stores just one copy of the duplicated e-mail, dramatically reducing the storage requirements. The Resource Manager can also use IBM Tivoli Storage Manager for the storage management of its data. Using Tivoli Storage Manager, the Resource Manager can offload data to other media such as DR550, EMC Centera, and a large number of tape devices.

CommonStore for Lotus Domino and CommonStore for Exchange Server

CommonStore for Lotus Domino and CommonStore for Exchange Server archive and retrieve data between their respective e-mail systems and Content Manager. When using eMail Search, it is common to implement journaling within the e-mail system. Journaling is used to ensure the capture of every e-mail that passes through the e-mail system before it is delivered to the recipients. The captured e-mails are then pushed into Content Manager using CommonStore. For more information about journaling, see Chapter 6, “Journaling” on page 85.

For an overview of CommonStore for Lotus Domino and CommonStore for Exchange Server, see 1.2, “IBM CommonStore for Lotus Domino” on page 3, and 1.3, “IBM CommonStore for Exchange Server” on page 8.

eMail Search server and client

eMail Search server runs under WebSphere Application Server. eMail Search client is a Web application that offers a search interface for e-mails stored within Content Manager. The integration with Content Manager is performed using the Content Manager application programming interfaces (APIs).

eMail Search server also has integration with Lotus Notes and Microsoft Exchange. The integration allows users of eMail Search to export data out of Content Manager into their original format.

7.4 eMail Search installation and configuration

eMail Search installation comprises a number of steps that you have to perform. We describe from a high level the tasks that you should perform and the order in which to perform them.

7.4.1 eMail Search installation prerequisites

Before you install eMail Search, it is important to make sure that you have all of the prerequisites installed and correctly configured. The following section provides a list of the prerequisites and version levels at the time of writing this IBM Redbook. Understand the prerequisite requirements. Check your system to ensure that you have the latest version and fix packs required for the software. Check the readme file that comes with the installation package for any amendments before installing.

Content Manager and eMail Search:

- IBM Content Manager V8.3 with fix pack 3 plus test fix 1 of Content Manager V8.3 with fix pack 3 or later
- IBM DB2 Universal Database™ Enterprise Server Edition V8.2 with fix pack 12 or later
- IBM DB2 Net Search Extender V8.2 (NSE V8.2) with fix pack 9 or later
- IBM WebSphere Application Server V5.1.1.3 or V6.0 or later (Important: WebSphere Network Deployment is not supported.)
- IBM Information Integrator for Content Manager V8.3 with fix pack 3 plus test fix 1 of Content Manager V8.3 with fix pack 3 or later

Domino:

- IBM CommonStore for Lotus Domino V8.3.1 or later
- IBM Lotus Domino 6.5.4

Exchange:

- IBM CommonStore for Microsoft Exchange V8.3.1 or later
- Microsoft Exchange Server 2000 or later
- Microsoft Outlook 2003 or later

Web browser

- Microsoft Internet Explorer® 6.0 or later

The test fix is available at the CommonStore download site:

`ftp://ftp.software.ibm.com/software/commonstore`

7.4.2 eMail Search installation planning

eMail Search consists of a number of components, which you have to consider when planning on how to deploy the solution. For test and demonstration environments, it is possible to deploy all of the components on the same physical machine. For a production environment, we recommend that you split the components across multiple machines. You should work with your IBM representatives to perform a sizing exercise (as discussed in Chapter 4, “Sizing” on page 45) when planning the architecture and sizing (see also Chapter 3, “Solution design” on page 27).

Before you install eMail Search component, you should have a fully operational CommonStore environment. Configure CommonStore to make sure that you are getting the maximum throughput of archived e-mails from the journal. You should try to obtain a throughput of approximately four e-mails per second per CommonStore instance.

Tune the Content Manger to make sure that you are getting the maximum performance. This includes tuning IBM HTTP Server, IBM WebSphere Application Server, DB2, and the operating system. See *Performance Tuning for Content Manager*, SG24-6949.

7.4.3 eMail Search installation and configuration

Detailed documentation is available that describes eMail Search installation and configuration. We do *not* repeat everything that is included in the documentation. In this section, we emphasize on a sequence of the steps that you have to perform for eMail Search installation and configuration based on our experience. Some of the steps or substeps are interchangeable. We outline the sequence as best practices, but this is not the only way to set up the system. As eMail Search continues to evolve, always check the latest documentation for any changes in the actual installation and configuration steps.

Based on our experience, we recommend the following eMail Search installation and configuration sequence:

1. Install Content Manager Version 8.3 with fix pack 3 plus test fix.

   Validate that Content Manager can import and retrieve a document.

2. Install CommonStore for Lotus Domino or Exchange Server Version 8.3.1.

   Validate that CommonStore can archive and retrieve an e-mail.

3. Install eMail Search.

   Validate that the application is installed properly.

4. Configure the system.

   Validate that you can archive and search e-mails.

5. Set up for and perform high-speed document indexing.

Step 1: Content Manager installation

The Content Manager installation includes the following steps:

1. Install DB2.

   > **Important:** This has to be *64-bit on AIX* whenever possible. It must be defined as Unicode Transformation Format-8 (UTF-8) for high performance.

2. Install DB2 Net Search Extender.

   Configure CommonStore Text Search User Exit.

3. Install WebSphere Application Server.

   > **Important:** Make sure that you install the relevant fix packs.

4. Install Content Manager on the server.

   > **Important:**
   >
   > - Choose a custom installation and select the *unicode* installation.
   > - Make sure that Content Manager has the latest fix packs installed. Make sure that you check the install.log file in the Content Manager fix pack directory as the fix pack update might show that it is successful, however, there can still be errors.

5. Install the Content Manager client for testing.

6. Test the installation by importing some documents into Content Manager and then retrieve them from the Content Manager client.

Step 2: CommonStore installation

The CommonStore installation includes the following steps:

1. Install CommonStore for Lotus Domino or CommonStore for Exchange Server.

2. Configure the system according to the product manual.

3. Validate that CommonStore works by archiving an e-mail, and then retrieve the archived e-mail for viewing.
4. Configure the system to use mail journaling database.
5. Validate that journaling work by sending an e-mail, make sure that the e-mail is journaled and that the CommonStore is triggered to archive the e-mail.
6. Make sure that the system is optimized such that you can archive approximately three to four e-mails per second per CommonStore server (instance).

Step 3: eMail Search installation

eMail Search installation includes the following steps:

1. Ensure that the following prerequisites are installed on eMail Search server and client machine:
   - For Domino on Windows environment, Notes client

     For Domino on AIX environment, Domino server

     For Microsoft environment, Microsoft outlook
   - IBM WebSphere Application Server
   - DB2 administration client (runtime client is also acceptable)
   - Information Integrator for Content

   > **Important:** Detailed steps on how to install these prerequisites and what options to select when performing installation are provided in section 3.1 and section 3.2 of eMail Search documentation. Perform the detailed steps described in the documentation.

2. Change DB2 setting for the Library Server database. Note that this step can be done earlier. We specify the step here because often you might start with a system that already has Content Manager installed and you might bypass the Content Manager installation step and can miss this important step if we put it in that section.

> **Important:** To ensure that the system performs well, set the following for the Library Server database:
>
> ```
> db2 update db cgf for icmnlsdb using
>         APP_CTL_HEAP_SZ     30000
>         APPLHEAPSZ          60000
>         PCKCACHESZ          16384
>         MAXAPPLS            100
>         AVG_APPLS           20
>
> db2 update dbm cfg using
>         SHEAPTHRES          20000
>         MAXAGENTS           500
>         NUM_POOLAGENTS      500
>         MAX_CONNECTIONS     500
>         FENCED_POOL         100
> ```

3. Configure NSE. Again, you can perform this step earlier after NSE is installed. We specify the step here because you might start with a system that already has NSE installed and you might bypass the NSE installation step and miss this important step if we place it in that section.

> **Important:** Make sure that NSE configuration is set up as follows:
>
> ```
> BlockMode=ON
> BlockSize=LARGE
> RespectCase=OFF
> TreatNumbersAsWords=FALSE
> SeparateParagraphs=OFF
> SeparateSentences=TRUE
> ```
>
> The bold text given in the previous section is important. Make sure that you do *not* set BlockMode to DYNAMIC.
>
> NSE configuration can be done in .../sqllib/db2ext/cteixcfg.ini.

4. Before eMail Search installation, perform the following tasks:

   – Check that the environment variables are configured in install.bat or install.sh.

> **Important:** Make sure that the environment variables used in install.bat exist in your system. If they do not exist or the names do not match, create these variables in your environment. At the time of the writing, these environment variables are:
>
> ```
> IBMCMROOT
> DB2PATH
> ```

- For AIX, ensure that your environment is set up for DB2. You can check this by running the **db2profile** command.
- Check to see if the following item types already exist:
  - For CSLD: CSLDMail
  - For CSX: CSXMail

> **Important:** If you configured CommonStore using the default, you will have a conflict. If this is a new system, rename the item type in the Content Manager to something else.

If the item type already exists and you do not want to rename it, during eMail Search installation, you have to select not to create the item type during the installation. You have to manually create the item type as per the instructions later.

> **Recommendation:** Rename or delete the existing item type if possible and let eMail Search installation create the item type with all the proper setup. This is a much simpler installation process.

5. Record the values that you use for eMail Search installation. See Table 7-1.

Table 7-1 System values for installation

Configuration parameter	**Sample input**
DB2 installation directory	C:\IBM\SQLLIB
Content Manager installation directory	C:\IBM\db2cmv8
WebSphere Application Server installation directory	C:\WebSphere\Appserver
Cataloged Library Server database name	icmnlsdb
Library Server schema name	icmadmin
Library Server database connection ID	icmadmin

Configuration parameter	Sample input
Database connection password	
Mail server host name	
Mail server port	
Mail server user name	
Mail server password	
Mail server name	
eMail Search mail database (optional for native viewer only)	
Host and port of the Lightweight Directory Access Protocol (LDAP) server	
LDAP user	
LDAP password	
Search base for Public Address Book (NAB) lookups	

6. Install eMail Search by running the following command:

   – For Windows: **`install.bat`**
   – For AIX: **`install.sh`**

   The documentation provided along with eMail Search describes in detail how to install eMail Search server and client. For details, see section 4.1 of the documentation.

7. Perform additional steps after eMail Search installation, as described in section 4.1.1 of eMail Search documentation.

   You also can enable logging for eMail Search client by modifying ems-server-config.xml, as described in the documentation, section 4.1.3.

8. If you did not use the default item type name, create them using the Data Definition Language (DDL) generated from eMail Search or create it manually.

   By default, the DDL and the command to run it can be found under the path defined by the environment variable IBMCMROOT:

   – For Windows: %IBMCMROOT%\cmgmt\ls\<*database name*>\UserDDL
   – For AIX: /home/icmcmadm/cmgmt/ls/<*database name*>/UserDDL

Section 4.1.11 of eMail Search documentation describes in details the item type specification. See this section (and also 4.1.12) for information about how you can create it manually or what is set up for the item type.

9. Attributes indexing: Regardless of whether the item type is created automatically or manually, double check to ensure that the three attributes mentioned in the following note are indexed.

> **Important:** Create DB2 index for the following attributes of the CSLDMail or CSXMail item type:
>
> - CSCDISIS
> - CSCRISIS
> - CSPostedDate (Domino) or CSEmailDate (for Exchange Server)
>
> If you do not index these attributes, CommonStore performance decreases when the volume of e-mails increases.
>
> The ICMDeleteHold attribute is required for native holds. This attribute is configured by the installation program and must be called *ICMDeleteHold* if you create it manually.

10. Configure CommonStore Library Server for full text search.

    Depending on your installation (CSLD or CSX), instructions about how to do this can be found in:

    – *CommonStore for Exchange Server V8.3.1 Administration & User's Guide*, SH12-6741

    – *CommonStore for Lotus Domino V8.3.1 Administrators and Programmer's Guide*, SH12-6742

11. Configure CommonStore to point to the CSLDMail or CSXMail item type in the archint.ini, if it is not already done. If you used a non-default item type, configure CommonStore to point to that item type.

    Depending on your installation (CSLD or CSX), instructions about how to do this can be found in:

    – *CommonStore for Exchange Server V8.3.1 Administration & User's Guide*, SH12-6741

    – *CommonStore for Lotus Domino V8.3.1 Administrators and Programmer's Guide*, SH12-6742

> **Important:** When you set up CommonStore to archive e-mail, two settings must be set for eMail Search to work properly:
>
> - When you set up item type in Content Manager, use BUNDLED data mode in the CommonStore server configuration profile, archint.ini. For example:
>
> ```
> ARCHIVE A1
>   STORAGETYPE     CM
>   ITEM_TYPE       CSLDMail
>   LIBSERVER       icmnlsdb
>   CMUSER          icmadmin
>   ARCHIVETYPE     BUNDLED
>   SISCHILDNAME    SISChild
> ```
>
> - The archiving type must be Entire. The entire document, not just the attachments, must be archived.
>
> For more information, see Appendix A of eMail Search documentation.

12. Update the DB2 full text index.

> **Important:** Check for any errors here.

13. Start eMail Search server and eMail Search client.
    - To start and stop eMail Search client:
        - For Windows: Run startEMSApp.bat and stopEMSApp.bat.
        - For AIX: Run startEMSApp.sh and stopEMSApp.sh.
    - To start and stop eMail Search server:
        - For Windows: Run startEMSServer.bat and stopEMSServer.bat.
        - For AIX: Run startEMSServer.sh and stopEMSServer.sh.

      When you run AIX commands, you must run with a non-root user ID.

      Check server output file for any errors. By default, it is ems.log for Windows and nohup.out for AIX.
14. Validate that the installation works by launching eMail Server application and perform a full text search. Use the Web preview to view the e-mails.

    At this stage, you cannot perform a search based on date or user name. You cannot view e-mail using Webmail or native Notes because they have not yet been configured.

Step 4: Configuration of the system

The configuration includes performing the following steps:

- Configure group expansion.
- Set up LDAP.
- Set up Notes password.
- Configure export.
- Set up native viewing.
- Set up single sign-on (optional).

Configuring group expansion

Group expansion is used to expand the destination or receiving group e-mail address to individual addresses internally before the e-mail is archived. Using group expansion, enable archived e-mail to be associated with the individual users within the group so that at a later time, e-mails can be searched based on the individual user rather than based on group only.

Tip: You can start using eMail Search system as is without configuring this step during the testing phases. This step can be deferred when you test the archiving and searching of the e-mails.

For detailed information about how to configure it, see section 4.4 of eMail Search documentation. Note that it is not our intention to repeat all the detailed steps already described in the guide. We emphasize on the sequence of the steps and what to watch out for during the installation and configuration. Use this section of the IBM Redbook in conjunction with the documentation.

Configuring LDAP

When LDAP is integrated with eMail Search client, users can perform lookup of names from the messaging address book for use in the *to* and *from* fields when searching e-mails.

Tip: Configure LDAP when you test whether eMail Search installation works. This step can be delayed until you have to search e-mails based on users looked up from the address.

- Microsoft Exchange

  Microsoft Exchange uses Microsoft Active Directory for name and address resolution. eMail Search client integrates with Active Directory using LDAP. Example 7-1 shows a sample of how we configure eMail Search for use with our Microsoft Exchange environment. We add lines to the ems-config.xml file, as shown in Example 7-1.

Example 7-1 ems-config.xml update

```
<ldap-server>
<host>cm8.domain.demo.com</host>
-<user>administrator@domain.demo.com</user>
<password>AdminPassword</password>
<search-base>cn=users,dc=domain,dc=demo,dc=com</search-base>
<max-results>100</max-results>
</ldap-server>
```

- Lotus Domino

  Lotus Domino uses the Lotus Domino Directory for name and address resolution. eMail Search client integrates with the Lotus Domino Directory using LDAP. For the integration to work, enable LDAP on a Domino server. After LDAP is enabled on the Domino server, eMail Search has to point to this Domino server for LDAP authentication.

  For more details, see section 4.1.9 of eMail Search documentation.

Setting up the Notes password

The purpose of this step is to bypass the Notes password prompt when viewing e-mail in the native Notes format. See section 4.1.7 of eMail Search documentation for details.

> **Tip:** You do not have to set this up to test or use your eMail Search system. You can perform this step at a later time.

Configuring export

The purpose of this step is to configure eMail Search server for export. See section 4.1.6 of the documentation for details.

> **Tip:** When you make changes to ems-server-config.xml, be careful not to add unnecessary spaces. It might introduce errors.
>
> You do not have to set this up to test or use your eMail Search system for archiving and searching e-mails. You can perform this step at a later time.

Native viewing

Users of eMail Search can use native viewing to view archived content in its native format, either the Lotus Notes client or Microsoft Outlook. We describe the processes for native viewing using both the Microsoft Exchange and Lotus Domino e-mail environments.

> **Tip:** You do not have to configure this to test and use your eMail Search system. You can do this at a later time.

- Microsoft Exchange native viewing

  When you implement eMail Search in a Microsoft Exchange environment, you do *not* have to perform additional setup to enable native viewing.

  From a user perspective, they perform a search and they are presented with a list of results. The result list contains two icons next to each result, the first is an e-mail icon, which users can use to preview the e-mail. The second is a Microsoft Outlook symbol, which the user can click to launch the e-mail in Microsoft Outlook.

  The native viewing does not require any configuration as when the e-mails are served back to the user through eMail Search using Microsoft Internet Explorer. They are returned in their original .msg format with the correct mime type. Microsoft Internet Explorer forces the e-mail to be displayed through Microsoft Outlook provided it is installed and configured.

- Lotus Notes native viewing

  When you implement eMail Search in a Lotus Domino environment, you *have* to perform additional setup of eMail Search.

  From a user perspective, they perform a search and are presented with a list of results. The result list contains three icons next to each result. The first is an e-mail icon, which the user can use to preview the e-mail. The second is a Lotus Notes symbol, which the user can click to launch the e-mail in the Lotus Notes client. The third is a Lotus Domino Web access symbol, which the user can click to launch the e-mail in a Lotus Domino Web access format.

  For additional details, see section 4.1.8 of eMail Search documentation.

Step 5: Setting up for and performing high-speed indexing

The setting up for and performing high-speed indexing includes the following steps:

1. GenTie2 and ICMFeatchFile should be copied to ../sqllib/function on AIX (using binary files), or ..\sqllib\function on Windows. Run chmod 755 on AIX to grant access.
2. Connect to the database and run the following command:

   ```
   db2 set current schema icmadmin
   db2 -tvf GenTie2.ddl
   db2 bind GenTie2.bnd
   db2 -tvf ICMFetchFile.ddl
   ```

   > **Note:** Whenever GenTie2 is updated, you must run the **`db2 bind`** command again.

3. Ensure that the NSE index and work files are on the same file system. The file system should be up to four times as large as the size of index.

   If they are not on the same file system, use **`db2text alter`** command to move the index files or the work files.

4. Prepare the existing NSE index by running the following command:
   - For Windows:

     ```
     fetchcontentsetup.bat <item type name> <LOG or RUN>
     ```

   - For AIX:

     ```
     fetchcontentsetup.sh <item type name> <LOG or RUN>
     ```

   Here <*item type name*> by default is CSLDMail for CSLD system and CSXMail for CSX system. Use LOG to display DB2 statements that are run. Use RUN to run the statement.

   For example on Windows, CSLD environment, perform a LOG and then a RUN:

   ```
   fetchcontentsetup.bat CSLDMail LOG
   fetchcontentsetup.bat CSLDMail RUN
   ```

5. To index the documents, perform the following command:
   - For Windows:

     ```
     indexdocuments.bat <item type name> MAXRUNTIME INTERVAL
     ```

   - For AIX:

     ```
     indexdocuments.sh <item type name> MAXRUNTIME INTERVAL
     ```

MAXRUNTIME specifies the number of hours for indexdocuments to run. If this is set to 0, only one iteration will run.

INTERVAL specifies how often to run indexdocuments. If this is set to 0, processing will continue with no break until all the documents are indexed.

The output of the document indexing process is stored in <*item type name*>.log in the same directory where you run the indexdocument task.

6. After you set the system to index the documents, you have to monitor the results. From time to time, you have to control the index and retry if there are errors. eMail Search documentation provides details about these steps. See sections 4.2.4, 4.2.5, and 4.2.7 for details. Section 4.2.7 also provides a functional overview of eMail Search. We recommend that you read it to get a better understanding of what the system does when indexing the document.

   Other sections in eMail Search documentation are equally important for you to read and understand. This include section 4.2.8, “Planning for Recovery”, and section 4.2.9, “Reporting”.

   > **Note:** Back up the NSE full text index on a regular basis. Make sure that you read section 4.3 on NSE backup in eMail Search documentation.

In conclusion, we outlined the steps that you have to perform for eMail Search installation and configuration. You must use the materials here in conjunction with eMail Search documentation to properly install and configure the system. There is also a troubleshooting section at the end of the documentation that helps you to resolve problems that are related to eMail Search.

8

Maintenance and troubleshooting

This chapter describes the maintenance tasks that are necessary to keep a CommonStore solution up and running smoothly and efficiently. To ensure a successful solution setup and continued performance when the solution is deployed, there should be an effective maintenance, backup, and recovery plan for the CommonStore solution. Incorporating the planning and implementation of maintenance, backup, and recovery into the overall deployment of the solution is important and should not be missed. We outline the tasks that are necessary to maintain an efficient solution.

We cover the following topics in this chapter:

- Monitoring and maintenance
- Backup and recovery
- Maintaining CommonStore solution integrity
- CommonStore troubleshooting hints and tips
- Purged e-mail management (for CSLD only)
- Maintenance review

8.1 Monitoring and maintenance

It is best practice to perform routine monitoring and maintenance of the system to ensure the smooth operation of a CommonStore solution. This includes monitoring and maintaining the database size and removing the old trace and log files so that they do not become too large over time.

> **Note:** Some of the information provided in this chapter are extracted from the existing product manuals and IBM Redbooks. We outline the areas that you have to be concerned with. For more information, see these documentations mentioned at the end of this section.

A CommonStore solution contains several key components:

- A mail server, such as Lotus Domino or Microsoft Exchange server
- CommonStore server
- Back-end archive repository server, Content Manager (and other servers including OnDemand and Tivoli Storage Manager, which we do not address in this IBM Redbook)

Each component has maintenance tasks that should be performed regularly. This practice provides the best possible performance from the solution over time, and potentially avoids problems that can manifest themselves as system-endangering situations.

8.1.1 CommonStore server maintenance

For the CommonStore server in your solution, perform the following tasks on regular basis:

- Monitor and maintain the following log files on a *daily* basis:
  - CommonStore server logs (ai<*yyyymmdd*>.log)
  - CommonStore server error log (csserror.log)
  - Content Manager Version 8 error log (cm8errors.log)
  - CommonStore tasks and crawlers have their own logs and traces

  The log and trace files are written to the directory that is determined by the INSTANCEPATH parameter in the server configuration file, archint.ini. For example, the default CommonStore for Lotus Domino (CSLD) directory on Windows system is C:\IBM\CSLD\server\instance01. For CSX, the location of Task traces and logs is configured in the System Manager. For CSLD, the location of the Task traces and logs is configured in the configuration database.

Monitor the content of these files. Check the error logs and search for errors within them. Make sure that you check all the errors. Delete the ai logs that you no longer require or move them to some other disk.

- Monitor and maintain the trace files as required:

  - CommonStore server trace file (archint.trc)
  - CommonStore server startup trace file (startup.trace)
  - Content Manager Version 8 agent trace file (cmtrace.trc)

  Delete any trace files that you no longer require or move them to another disk.

  Turn tracing on only when there is an issue in your system. When tracing is enabled, the system does not disable it automatically. You must manually disable tracing. The logs that are generated by tracing are not automatically deleted either. After you resolve your issues, delete the trace files. If you want to save them for future reference, remember to monitor these trace files and delete them periodically as required or move them to another disk.

  Ensure that the size of the startup.trace file is within acceptable limit because there is no size limit for this file.

- For CSX, monitor the Operation log file, which contains similar information as reported by the CSLD report logging feature (mentioned below). Refer to CSX product manual for more details.

- Monitor the crawler log files for problems encountered in the crawler. They can also be used to figure out, why a specific mailbox is not crawled and you can also follow the decisions taken for every single message.

- For CSLD, run the report logging feature to gather information about the health of the system.

  This is an effective new tool. It logs all the CommonStore operations such as archive, delete, update, retrieve, and searches. Use this data to monitor e-mail archiving volume and the health of the CommonStore system.

  Monitor and review the *XXXXXXXX*.out file. This file shows any errors that are encountered during retrieval and conversion, and contains a line for each of the 1,000 documents that are retrieved and converted. This file is located in the same directory as where you ran the indexdocuments script.

  > **Important:** In the *XXXXXXXX*.out file name, *XXXXXXXX* represents the name of the item type that the CommonStore server uses to archive e-mails.

  This is configurable using parameters in the emsutil.properties file.

  It is always a good practice to review the logs file to determine the health of the system.

You can notify the CSLD administrators by selecting the users to notify in case of severe errors (for example, full disks, undefined archives, or lost network connections). If a severe error occurs, the listed users receive an e-mail notification including the job document of the job processed at the time of the error.

8.1.2 Content Manager maintenance

Content Manager is one of the back-end archive repository server for the CommonStore solution and is the focal point in this IBM Redbook.

Perform the following tasks regularly for the Content Manager back-end server:

- Optimizing the Content Manager server databases (using runstat and reorg)
- Monitoring LBOSDATA directory size
- Managing staging directory space
- Removing entries from the events table
- Managing Resource Manager utilities and services
- Replacing or re-partitioning a hard disk
- Backing up Library Server and Resource Manager database logs
- If you are not immediately archiving e-mails, backing up the directory that contains the untransferred e-mails; the default directory is C:\IBM\CSLD\server\instance01\transfer

We do not repeat the details of these tasks in this IBM Redbook because they are documented in the product publication manuals and other IBM Redbooks. We only outline the tasks to remind you that planning and performing of these tasks should be part of a CommonStore solution setup and is one of the best practices to ensure the solution's functioning as expected over time.

For more specific details about how and when to do the maintenance tasks, we recommend that you read the associated IBM Redbooks and the product manuals in conjunction with this IBM Redbook:

- *Performance Tuning for Content Manager*, SG24-6949
- *Content Manager Backup/Recovery and High Availability: Strategies, Options, and Procedures*, SG24-7063
- *Content Manager Enterprise Edition V8.3: System Administration Guide*, SC27-1335

- *CommonStore for Lotus Domino V8.3.1 Administrators and Programmer's Guide*, SH12-6742
- *CommonStore for Exchange Server V8.3.1 Administration & User's Guide*, SH12-6741

8.2 Backup and recovery

It is a good practice to periodically test your backup and restore strategy. One of the worst scenarios that can happen to your company is your server crashes, you lose data, and then you realize that your backup device is not working properly and the restore procedure fails.

When you develop a recovery procedure, make sure that the procedure is as simple as possible. You must get your server up and running quickly when the server is down.

We recommend that you perform full online backups every day whenever possible. If this is not possible, we recommend that you perform a full offline backup once a week and incremental backups on a daily basis during the rest of the week. If your server crashes in the middle of the week, you can use the full backup from the previous week and apply the incremental backups.

To help you to prepare for backup and recovery, we address the following topics in this section:

- Operating system backup
- Database backup

There are other files that you have to consider backing up. For Lotus Domino, these include Notes e-mail databases and mail journaling databases. For Exchange Server, these include mailboxes and public folders. Consult the product manuals for the backups.

8.2.1 Operating system backup

Depending on how bad the system crash is, you might have to set up the entire CommonStore system from the beginning. A CommonStore system has several components that can take time to reinstall, and you have to remember and apply the changes that you have made to the system. To speed up the potential recovery scenario, you should have a backup of the current operating system.

Back up the operating system regularly. We recommend that you do this backup on a biweekly basis. Whenever you make any changes to the system, such as apply updates, patches, and fixes, it is important to perform a backup before and after the changes.

> **Note:** Sometimes backing up the operating system is not enough. In a Windows network, the required information goes beyond the scope of an individual machine. For example, when using CSX, almost all configuration information is stored in Active Directory. If this information is corrupted, all configuration data is lost. Understand where your information is stored and back them up regularly.

8.2.2 Database backup

There are different types of backups. A combination of several types of backup methods is often employed for a complete backup strategy.

Database backup methodology

The factors in deciding which type of the backups to use and how often to perform the backup include:

- Ease of restoration

  Always keep restoration in mind when deciding the backup policy. A backup policy that is too complicated to restore might take a long time to complete. It may also introduces more errors or failure point during restoration. Comparing all the backup methods, a full offline backup is the easiest and most reliable method for restoration, because it does not rely on any logs.

- Backup window and system available time

  Some systems have to run 24x7. There might not be enough time to perform an offline full backup. In this scenario, an online full backup might become the only option. For the scenario where users only have to access the system during usual business hours, it is feasible to perform offline backup, even every day.

- Size of database

  Depending on the size of the database, it might be more viable to back up the database using incremental instead of full backup if the database size is huge. If the frequency of online or offline full backup decreases, there might be side effects. First of all, the size of each of the incremental backup increases because incremental backup is based on the last full backup, and there can be more and more changes since the last full backup. Secondly, more table spaces might have changes, and therefore more backup images might be generated as the time between the last full backup becomes further apart. Thirdly, the time taken to complete the incremental backup increases.

- Purpose of backup

  If a backup is performed because of an impending migration or upgrade, then a baseline of the data loaded has to be identified. Because there is a higher chance to perform restoration than is usual in this case, ease of backup and confidence of the backup become important. Due to the purpose of this type of backup, it is best practice to perform a full offline backup.

In general, if the backup window allows, perform offline full backup every day. If the system has to be available most of the time, then a combination of offline and online full backup is fine. If you do not have the luxury to take the system down at any given time, then perform a full online backup at least once a week, with incremental backup in between. This is to ensure that the incremental backups do not grow so much that they are unmanageable. Again, with the last option, both the restoration time and complexity increase.

Preparing an offline backup

Before creating an offline backup, it is necessary to stop all the activities on the CommonStore server and the Content Manager system.

Shut down the following components before an offline backup:

- Archpro
- Task agent (CSLD)
- Crawler agent (CSLD)
- CSX Task (CSX)
- Search Server (CSX)
- eMail Search Server
- DB2 Net Search Extender

To ensure proper system shutdown, we recommend that you shut down the services and agents in the following order:

- For CSLD:

  a. Task agent
  b. Crawler agent
  c. Archpro
  d. eMail Search Server
  e. B2 Net Search Extender

- For CSX:

  a. CSX Search Server (if not controlled by archpro)
  b. CSX Task

c. Archpro

   i. HTTP Dispatcher
   ii. CM8 agent
   iii. CSX Search Server (if controlled by archpro)

d. eMail Search Server
e. DB2 Net Search Extender

8.3 Maintaining CommonStore solution integrity

A CommonStore system contains several databases and log files that you must back up at the same point in time to maintain the integrity of a healthy system. The databases not only store the indexes for all the CommonStore data, they also contain the Content Manager definitions. To ensure that your backup strategy is complete and sound, we include a list of these files and databases that you must back up:

- Job database (for CSLD)
- Configuration databases (for CSLD)
- Configuration data in Active Directory (for CSX)
- Job folders (for CSX)
- Configuration folders (for CSX)
- Content Manager Library Server and Resource Manager databases
- Full text index (NSE) database
- eMail Search databases
  - EmailPurge.nsf (for CSLD)
  - EmailPurgeLog.nsf (for CSLD)
  - groupexpansionlog.nsf if journaling is enabled on the Notes Domino Server (for CSLD)
- E-mails

Job folder for CSX

The CSX client enables Outlook users to process their messages using CommonStore functions. Every time a user starts an action, a CommonStore job is created in the *job folder* of a CSX task instance. This is the instance associated with the user's mailbox. A document in the job folder represents a CommonStore job. The job document holds all the parameters that the CSX task requires to perform the requested operation. Each CSX task instance polls exactly one job folder. Thus, jobs from Outlook clients connected with the Exchange servers that are assigned to a particular instance are all posted to the same job folder. Jobs in a job folder go through different states, which reflect the processing progress. When the operation is completed, the jobs are deleted by default. If an error occurs when processing a particular job, the job remains in the job folder. You can track down the errors in question by reviewing the csserror log file. The file contains information about the cause of the errors. When the error is corrected and the archiving process is restated, you can remove the error job from the job folder. If you leave the error entries as is, we recommend that you periodically delete them to avoid cluttering.

Configuration folder for CSX

CSX client components read the configuration data provided by the administrator. When configuring a CSX task instance, the administrator assigns Exchange servers or journal recipient mailboxes to the instance and specifies a job folder for it. For each task instance, this information is written to a job folder mapping document in the configuration folder. This way, the CSX client components can determine the correct job folder to post jobs to.

CSLD job database

The CSLD job database is a Lotus Notes database in which all client requests (archiving, retrieval, searching, and viewing) are collected before they are picked up by a CSLD task. The CSLD job database is located on a Lotus Domino server.

CSLD configuration database

The CSLD configuration database is a Lotus Notes database that holds configuration documents for CSLD. For example, it contains the configuration documents for the CSLD task and the policies required for automatic archiving. The CSLD configuration database is located on a Lotus Domino server.

CM Library Server and Resource Manager databases

Content Manager Library Server and Resource Manager databases store archived e-mails. Back up these databases on a regular basis and at the same time as the CommonStore job databases.

Full-text index (NSE) database

Back up the files created by NSE to store the full-text indexes on a regular basis. NSE files might be corrupted if there is a system failure during a very small window frame when files are being copied. Software errors might also cause incorrect content to be indexed.

eMail Search databases

For CSLD, there are three eMail Search related databases: EmailPurge.nsf, EmailPurgeLog.nsf, and Groupexpansionlog.nsf.

The *EmailPurge.nsf* database contains the agents that perform the e-mail purge process. The main view (Mail IDs by database) contains the local delete queue list. Ideally, this view becomes empty after the e-mail purge process runs to completion. If any of the e-mails are not deleted because a user mail database is moved out of the server, this view shows the record.

The *EmailPurgeLog.nsf* database contains all the log messages for the e-mail purge process. They are categorized by the names of the agents (Cataloger, Dispatcher, and Deleter) and are in the date ascending order. Each time the agent runs, it creates an entry in the log with information about how many documents it processed. Any errors that are encountered are reported to this log. In addition to the log, the Domino server's console also shows invocations of the agent.

Note: To validate that an e-mail is purged from the targeted user mail database during the e-mail purging process, open the user's Lotus Notes mail database (for example, user1.nsf) and verify that the mail is deleted. Also verify that there are no errors in the e-mail purge log database (EmailPurgeLog.nsf).

The *groupexpansionlog.nsf* database contains all the log messages for the Expand Group agent. This component consists of a single Lotus Notes database template containing an agent that performs the group expansion tasks intended to run before Lotus Notes journaling takes place. Lotus Notes journaling is a configuration option (turned off by default) on the Domino server that keeps a copy of e-mail that passes through the router in a separate database. For more information about journaling, see the Lotus Domino product manual.

8.4 CommonStore troubleshooting hints and tips

This section details the hints and tips about troubleshooting CommonStore for Lotus Domino and CommonStore for Exchange Server solutions.

Some of the problems that people face today are, for example, that the document archiving speed is too slow or not to one's expectations. Although everyone has different CommonStore setup and configuration, and it is difficult to tell you exactly what to do for each circumstances, we present some suggestions to help you troubleshoot your problem.

For specific problems, check the CommonStore support Web site for help.

- CommonStore for Lotus Domino product support:

  http://www.ibm.com/software/data/commonstore/lotus/support.html

- CommonStore for Exchange Server product support:

  http://www.ibm.com/software/data/commonstore/exchange/support.html

> **Tip:** When doing problem determination, we recommend to not start CommonStore as a service, but to start the individual components separately from command line.

8.4.1 Performance-related problems (archiving too slow)

An effective CommonStore solution, with standard central processing unit (CPU) processing speed and comparable memory, should be able to archive four e-mails per second per CommonStore server in a Lotus Domino environment, and three e-mails per second per server in an Exchange Server environment. With additions of CommonStore server, you can increase the number of e-mails to archive per second. With eMail Search, the system's e-mail archiving speed is dependent on the full-text indexing speed of e-mails. At the time of writing this IBM Redbook, with a tuned system, CommonStore eMail Search can full-text index anywhere between 25 to 35 e-mails per second. This number continues to increase as the development lab finds better ways to handle indexing and better ways to fine-tune the hardware and software in the system.

If the CommonStore solution that you set up is not archiving to the required speed, determine how many e-mails the system is indexing. What is your current e-mail archiving speed? What is the expected archiving speed? Is it within the capability of the CommonStore archiving speed as mentioned in the previous section? If not, there are improvements to be made.

For example, in a testing environment, a CommonStore system is set up to archive 500,000 e-mails. The system uses one 2.4 GHz, 2 GB RAM Windows 2000 machine for Content Manager V8.3 as the back-end, two 1.7 GHz, 1 GB RAM Windows 2000 server machines as CommonStore for Lotus Domino servers. It takes the system 20 days to archive all the e-mails. Is this an acceptable speed? Of course not. Let us examine the numbers.

In 20 days, there is a total of 1,728,000 seconds (20 days * 24 hours * 3600 seconds). The total number of documents that are archived during this period is 500,000. This roughly means that CommonStore system is taking approximately 3.46 seconds to archive one document (1,728,000 / 500,000).

Full-text indexing is not set up for the CommonStore system, therefore we do not discuss this in the analysis.

There is definitely a problem in this case. If we assume the standard speed of archiving four documents per server per second and the CommonStore system is set up optimally, the system should be able to handle archiving of all the documents within one day (24 hours * 3600 seconds * 4 doc/server * 2 server = 691,200 documents) or two days, based on a 16-hour archiving period.

Where is the problem? How we can troubleshoot the problem?

In general, there are several common areas that cause the archiving speed problem:

- *Configuration issues*: The most common one is missing index; for example, if Single-Instance Storage (SIS) is set and the CSCDISIS attribute is not set to be indexed, you can run into performance problem. See 8.4.2, “Missing index” on page 144 for more discussion about this topic.
- *Network throughput issue*: The connection between the CommonStore server and the Content Manager server is too slow and causes the problem.
- *Content Manager connection issue*: The connection between Library Server and Resource Manager is too slow and causes the problem.

  For example, you might have firewall issues; certain ports are closed; there are software changes; machines have been transferred from one place to another. This might also cause the connection problem between the CommonStore server and the Content Manager server.
- Too much load on the system.
- Others: For example, the network card has defects or the wireless area network (WAN) that is used is too slow.

The recommended troubleshooting sequence for performance-related issues (to see where the performance bottleneck is):

1. Check the -- ai*.log. Look for the time it takes the system to archive a document. How much time did the system spend in CommonStore and how much time is spent in Content Manager? See 8.4.3, “Analyzing the ai log” on page 144 for more discussion about this topic.
2. Check the CSLD log.
3. Check the archint.trace.
4. Check the cmtrace.trc.

The CommonStore logs and trace files help in troubleshooting. You can use different settings to specify what information to gather. These files are grouped under:

- CommonStore server log files
- CommonStore server trace files
- CSLD task and crawler logs and traces
- CSX task and crawler logs
- CSX Manager trace

> **Note:** If you are collecting data for the IBM support team, always collect all the log, trace, and trc files.

Because trace files tend to contain a lot of information related to the sequence of function calls, the input and output data to these functions, such information cannot always be easily understood and found. Searching for keywords such as error and fail often helps in identifying the errors. Also, remember to turn tracing off when the problem is resolved. Remove the trace files when they are no longer required.

The CommonStore server logs (including the Content Manager agent error log) are crucial in determining the failing component within the CommonStore solution. They gather information and report the health of the system.

These files include:

- CommonStore server log file (also known as ai logs)
- CommonStore server error log (csserror.log)
- Content Manager version 8 error log (cm8errors.log)

The CommonStore server console also displays the output of the server activities on the screen and you can use it to monitor server activities.

For more information about logs, trace files, and the new report logging feature, see Chapter 9, “Logs, traces, and report logging feature” on page 149.

8.4.2 Missing index

In most cases, the performance issue is caused by missing indexes.

For CommonStore for Lotus Domino:

- Make sure that the index on field Itemid is updated frequently.
- If you are using SIS archiving, make sure that the following indexes are created:
  - SIS: Create index for CRI CSCRISIS
  - SIS: Create index for CDI CSCDISIS

To optimize the performance as an item grows in size, you can index the CSLDOrigDB and CSLDOrigUser attributes if they are defined in your item types. Index these attributes in all item types used for CommonStore archiving.

8.4.3 Analyzing the ai log

To begin with common troubleshooting, start analyzing the ai*.log. Figure 8-1 shows the sample content of an ai log.

```
 Time;    RC;  Job;    Operation;   Archive;T_srv; T_ag;  PID;
IP address; ...
---------------------------------------------------------------------
-------------------------------------------------------
13:33:25;   0;   0;    ARCHIVE;     A1;0.658;0.638;03264;
;2006061313332450#0CC0.03A9;data;;;CSN;d:\CSLDexport\csn2.CSN;374924
8.0;
```

Figure 8-1 ai log sample content

The first line contains header for the data fields reported in the log entry. This log entry logs an ARCHIVE operation.

The first number in bold (0.658) shows the total time used to archive an e-mail (T_srv). The second number in bold (0.638) shows the amount of time that the operation spent within the archive agent (T_ag).

If the first number is much more than the second, there is a problem within CommonStore. Probably you have to increase the number of archive agents. If the second number is high (depending on the size of the e-mail, which is indicated by the third number in bold, 3749248.0), you might have to check the back-end (Content Manager) or the network.

8.4.4 Failure to archive e-mails

When your CommonStore system fails to archive e-mails, you usually go through the following set of actions:

- Check error description in the job document.
- If administrator notification is enabled, check the error message sent to the administrator.
- Check server and task log files for noticeable problem.
- If that does not help, turn on tracing.
- Set report on in the archint.ini files.
- Restart CommonStore.
- Monitor the activity of users and trace logs. Use them to identify problems.

8.5 Purged e-mail management (for CSLD only)

This section describes how to use the tools to clean up e-mails from the archived server when the stubbed e-mails are removed from the Lotus Notes user inbox.

Email Purge is a component that is used to delete archiving stubs from Lotus Notes. For example, an e-mail is archived with CommonStore. CommonStore places a stub in the e-mail that indicates that the e-mail is archived. Email Purge then goes ahead and deletes these e-mails.

Removing these purged e-mails is important to ensure the health of your system. To manage purged e-mails:

1. Configure Content Manager for Email Purge.
2. Delete e-mails from Content Manager using ExpireDocuments.

Configuring Content Manager for Email Purge

On the Content Manager Library Server, run the EmailPurgeInstall Java™ program. This program creates triggers on the Content Manager Library Server database tables to enable the Email Purge application to process items that have been deleted. This program also creates a new table in the Content Manager Library Server database called CS_STUB_CLEANUP that temporarily holds information about items that have been deleted.

Run the following command:

```
java com.ibm.cm.ems.emailpurge.EmailPurgeInstall <database> <userID>
<password> <childComponent>
```

Here <*database*> is the Content Manager Library Server database, <*userID*> is the Library Server administrator ID, <*password*> is the administrator ID's password, <*childComponent*> is the name of the child component in the Content Manager item type to be used by CommonStore for archives.

For example:

```
java com.ibm.cm.ems.emailpurge.EmailPurgeInstall icmnlsdb icmadmin
password CSLDSISCHILD1
```

> **Note:** To run this Java class, make sure the following DB2 Java library files are in the classpath:
>
> db2java.zip, db2jcc.jar, db2jcc_license_cu.jar, emailpurge.jar

Deleting e-mails using ExpireDocuments

Email Purge deletes Lotus Notes e-mail stubs that remain after the document is deleted from Content Manager. A tool is provided to delete documents from Content Manager. Because there are other methods to remove purged e-mails for Content Manager, using this tool is optional. For example, the system administrator can log on to the Content Manager solution and manually delete the purged e-mails or write an application that deletes the purged e-mails from the Content Manager system repository. The tool is a Java program called *ExpireDocuments*, and deletes documents that have an EXPIRATION attribute containing a date less than or equal to the current date.

Before you run this program, you must run the ExpireDocuments installation. This creates a trigger on the Library Server database so that the item cannot be deleted if the EXPIRATIONDATE attribute has not arrived. It also creates a trigger to prevent the EXPIRATIONDATE from being reduced.

To run the installation, use the following command:

```
java com.ibm.cm.ems.emailpurge.ExpireDocumentsInstall <database>
<userID> <password> <childcomponent>
```

For example:

```
java com.ibm.cm.ems.emailpurge.ExpireDocumentsInstall icmnlsdb icmadmin
password CSLDSISCHILD1
```

To check if the expired e-mails are deleted, check EmailpurgeLog.nsf file.

8.6 Maintenance review

We summarize a list of the maintenance tasks that you have to perform to keep your system working correctly and running smoothly:

- Ensure that the Resource Manager and Library Server databases are optimized regularly.
- Ensure that the Resource Manager utilities and services that you require in your environment are running and performing their jobs as expected.
- Regularly delete log files that you do not require any more, to prevent them taking up space unnecessarily. This is particularly important when you enable tracing, because the log files grow in size extremely quickly.
- Back up critical components on a regular basis, and test that these backups work on a regular basis by restoring your system to another machine.
- Make sure to turn off any tracing when they are not required. This improves the performance.

Planning for minimum risk CommonStore system failure

Having a good failure plan is important to minimize the risk of failure within a CommonStore system. After you have developed your failure plan, the next step is to step through your plan to minimize the effects of a failure. To reduce the impact of a failure that might impact your e-mail environment, we provide some recommendations that should be included in your site plan:

- Invest in backup resources.
- If you have multiple e-mail location, do not share any of the CommonStore databases between the locations.

  For example, consider a scenario where you have a Domino mail server in Los Angeles, U.S., and another server in Boeblingen, Germany. It is advisable for each location to have its own CSLD job database and CSLD configuration databases.
- Use multiple physical drives on the database server.
- Duplicate the CommonStore system with similar roles.

During the planning stage of the CommonStore solution, plan for maintenance, backup, and recovery as discussed. This helps in minimizing CommonStore system failure.

9

Logs, traces, and report logging feature

In this chapter, we describe the log and trace files that are generated by CommonStore for both CommonStore for Lotus Domino (CSLD) and CommonStore for Exchange Server (CSX). We also introduce the new report logging feature that is currently available for CSLD Version 8.3.1.

We cover the following topics in this chapter:

- Introducing CommonStore log files and traces
- CommonStore logs
- CommonStore trace files
- CSLD task and crawler logs
- CSX logs and traces
- Report logging feature (CSLD)

9.1 Introducing CommonStore log files and traces

To ensure that a CommonStore system runs smoothly and efficiently, it is important to monitor the CommonStore server activities using the log files that it produces. In addition, you have to understand the different trace options that are available to troubleshoot system problems.

The log and trace files take disk space. If you turn on traces, this might impact the system performance. When planning for a successful CommonStore solution, it is a good practice to properly set up these files and directories and put in place a plan to monitor and manage them regularly.

CommonStore provides the following server and Content Manager agent related logs:

- CommonStore server log (ai logs)
- CommonStore server error log (csserror.log)
- Content Manager Version 8 agent log (cs8errors.log)

CommonStore also provides various trace files for different components and operations:

- CommonStore server trace file (archint.trc)
- CommonStore server startup trace file (startup.trace)
- Content Manager Version 8 agent trace file (cmtrace.trc)

CommonStore tasks and crawlers have their own logs and traces:

- CSLD task and crawler logs and traces
- CSX task and crawler logs and trace files

New to CommonStore for Lotus Domino is a report logging feature that logs system operations in special report logs for later system activity analysis.

9.2 CommonStore logs

CommonStore log files are key in determining the failing component within a CommonStore system. They gather information and report the health of the system. In this section, we cover three logs that are produced by the CommonStore system: the CommonStore server log, the CommonStore error log, and the Content Manager Version 8 agent log.

The CommonStore server console also displays the output of the server activities on the screen and it is useful to monitor the server activities.

9.2.1 CommonStore server log (ai log)

CommonStore server log, also known as *ai log*, reports the progress of the ongoing work. It includes normal system operations and errors encountered by the system. The CommonStore server creates one ai log per day. The parameter setting in the archint.ini file determines whether to write to this log, and where to write the log file.

The naming convention for the ai log is ai<*yyyymmdd*>.log, where:

- *yyyy* indicates the year of the log file.
- *mm* indicates the month of the log file.
- *dd* indicates the day of the month of the log file.

To enable or disable the log file, add one of the following line to the archint.ini file:

- `LOG ON`
- `LOG OFF`

To specify the ai log path, add the following line to the archint.ini file:

```
LOGPATH <path>
```

Example, in archinit.ini, you can set:

```
LOG ON
LOGPATH C:\IBM\CSLD\Server\instance01\log
```

If the log path is not set, the default path is what is set for the <*INSTANCEPATH*> value. This value, by default is the subdirectory server\instance01 in the CommonStore installation directory.

Note: If the log path includes a blank, you must enclose the path with single quotation marks (‘). For example:

```
LOGPATH 'C:\IBM SOFTWARE\CSLD\Server\instance01'
```

Table 9-1 provides the ai log entry header definition.

Table 9-1 ai log entry header definition

Log data	Description
Time	Timestamp when the server finished processing a job
RC	Return code from each operation. Zero (0) indicates that the job is successfully processed; any other value indicates error
Job	The number of the job processed
Operation	Type of the job operation; possible values are: ARCHIVE, RETRIEVE, DELETE, QUERY, and UPDATE
Archive	Identifier of the archive used for processing
T_srv	Time spent in CommonStore server for the job execution (in seconds)
T_ag	Time spent in agent for the job execution (in seconds)
PID	Process ID of the agent that processed the job

Figure 9-1 show the sample output of an ai log file.

```
Log started at: Wed Jul 26 12:26:28 2006
    Time;    RC;  Job;    Operation;   Archive;T_srv; T_ag;  PID;     IP address;
...
-----------------------------------------------------------------------------------
----------------------------------------
12:26:28;     0;    1;      ARCHIVE;   MAILBOX;11.158;10.935;00000;
;A1001001A05L14A04806E87615;data;ea193ec7157d6f557679e59bb78570e6;c34a55e8025183076
f1f0e067860f817;text/xml;C:\IBM\CSLD\Server\instance01\temp\csn3.XML;97033.0;
12:33:15;     0;    2;     RETRIEVE;   MAILBOX;21.599;21.531;00000;
;;;ea193ec7157d6f557679e59bb78570e6;;;A1001001A05L14A04806E87615;ea193ec7157d6f5576
79e59bb78570e6;C:\IBM\CSLD\Server\instance01\temp\csn2;
Log stopped at: Wed Jul 26 12:35:20 2006
```

Figure 9-1 Sample ai log file content

9.2.2 CommonStore server error log (csserror.log)

CommonStore server error log, the csserror.log file, reports errors in the server and dependent agent processes. If there is any error, it is always logged. You can use the parameter setting in the archint.ini file to specify where to write the log file.

To specify the csserror.log file path, add the following line to the archint.ini file:

```
ERRORLOG_FILE <path>
```

For example:

```
ERRORLOG_FILE c:\csxins\inst1\log\csserror.log
```

The default error log file is <*INSTANCEPATH*> and csserror.log.

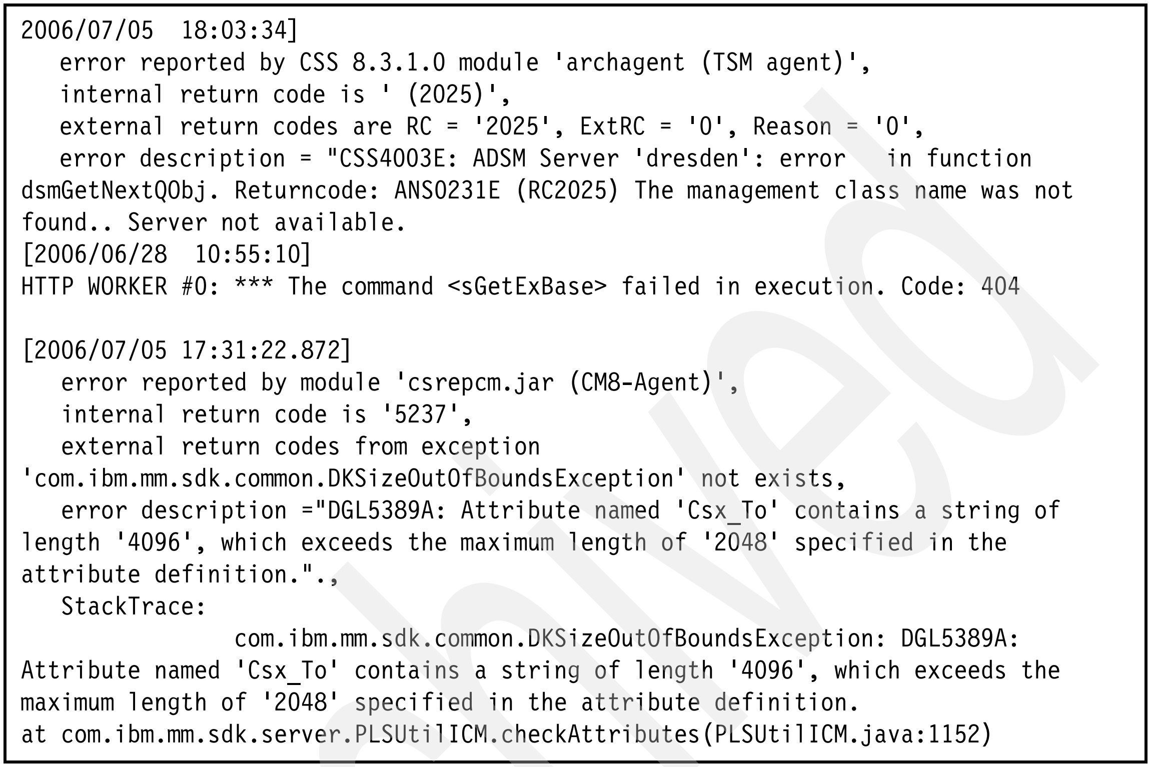
If your log file name or path has a blank character in the middle, remember to enclose it with single quotation marks (‘) when specifying it in the archint.ini file.

The entries in the error log file consists of one section per failed operation. Each section contains the following error information:

- Date and time when the error occurred.
- Component where the error occurred; this can be Content Manager, OnDemand, or Tivoli Storage Manager.
- Return code, extended return code if present, and reason code, if present.
- Error description obtained from the application programming interface (API) or generated by CommonStore.

Attention: The error log file grows without size limitation. It is a good practice to regularly monitor the file and clean it up as required.

Figure 9-2 shows the sample content of a csserror.log file. It shows a sample output of an error that occurred in a CommonStore for Exchange Server.



```
2006/07/05  18:03:34]
   error reported by CSS 8.3.1.0 module 'archagent (TSM agent)',
   internal return code is ' (2025)',
   external return codes are RC = '2025', ExtRC = '0', Reason = '0',
   error description = "CSS4003E: ADSM Server 'dresden': error   in function
dsmGetNextQObj. Returncode: ANS0231E (RC2025) The management class name was not
found.. Server not available.
[2006/06/28  10:55:10]
HTTP WORKER #0: *** The command <sGetExBase> failed in execution. Code: 404

[2006/07/05 17:31:22.872]
   error reported by module 'csrepcm.jar (CM8-Agent)',
   internal return code is '5237',
   external return codes from exception
'com.ibm.mm.sdk.common.DKSizeOutOfBoundsException' not exists,
   error description ="DGL5389A: Attribute named 'Csx_To' contains a string of
length '4096', which exceeds the maximum length of '2048' specified in the
attribute definition.".,
   StackTrace:
                com.ibm.mm.sdk.common.DKSizeOutOfBoundsException: DGL5389A:
Attribute named 'Csx_To' contains a string of length '4096', which exceeds the
maximum length of '2048' specified in the attribute definition.
at com.ibm.mm.sdk.server.PLSUtilICM.checkAttributes(PLSUtilICM.java:1152)
```

Figure 9-2 Sample csserror.log file content

9.2.3 Content Manager Version 8 agent error log (cm8errors.log)

Content Manager Version 8 agent error log, the cm8errors.log file, reports errors that occurred in the agent process when communicating with the Content Manager Version 8 back-end archive. This file is *always* written to the <*INSTANCEPATH*>.

Figure 9-3 shows a sample cm8errors.log file.

```
[2005/11/26 16:45:08.573]
                error reported by module 'csrepcm.jar (CM8-Agent)',
                internal return code is '5050',
                external return codes from exception 'DKUsageError':
ErrorId='5050', ErrorCode='7015', ErrorState='',
                error description ="DGL5050A: SQL error executing query.  XQPE
query: /CSLDProject[(@CSCDISIS = "4927a4f3e41a8bd06e1453d71f21b7f5")]. Return code:
7015.  Reason code: 0.  Extended return code: -204.  Extended reason code: 0.;
ICM7015: During an SQL operation in the library server, an unexpected error
occurred.  For details on the error, refer to the database document. (SERVER RC) :
7015".,
                StackTrace:
                com.ibm.mm.sdk.common.DKUsageError: DGL5050A: SQL error executing
query.  XQPE query: /CSLDProject[(@CSCDISIS = "4927a4f3e41a8bd06e1453d71f21b7f5")].
Return code: 7015.  Reason code: 0.  Extended return code: -204.  Extended reason
code: 0.;  ICM7015: During an SQL operation in the library server, an unexpected
error occurred.  For details on the error, refer to the database document. (SERVER
RC) : 7015
at com.ibm.mm.sdk.server.DKResultSetCursorICM.throwSQLException
(DKResultSetCursorICM.java:996)
at com.ibm.mm.sdk.server.DKResultSetCursorICM.prepareAndExecuteSp
(DKResultSetCursorICM.java:817)
at com.ibm.mm.sdk.server.DKResultSetCursorICM.prepareAndExecuteSp
(DKResultSetCursorICM.java:736)
at com.ibm.mm.sdk.server.DKResultSetCursorICM.open
(DKResultSetCursorICM.java:1083)
```

Figure 9-3 Sample cm8errors.log file

9.3 CommonStore trace files

Trace files help to trace problems by providing details of the operations. In this section, we cover the CommonStore server trace file (archint.trc), CommonStore server startup trace file (startup.trace), and Content Manager Version 8 agent trace file (cmtrace.trc).

Turning on traces slows down the system operation. By default, tracing is not turned on. Turn them on only for troubleshooting purpose. After you complete troubleshooting, turn off the traces. It is also a good practice to clean up the trace files periodically.

The default path used for trace files is <*INSTANCEPATH*>. The entry in archint.ini determines whether to write trace information. If there is any blank characters in the path or file name, use single quotation marks around it when specifying it in the archint.ini file.

9.3.1 CommonStore server trace file (archint.trace)

CommonStore server trace file, the archint.trace file, contains tracing information for all server components operation except the Content Manager Version 8 agent.

The archint.trace file includes the following tracing information:

- Timestamp
- Process ID
- Module name
- Trace message

To enable or disable tracing, add one of the following lines to archint.ini:

- `TRACE ON`
- `TRACE OFF`

To specify the trace file name, add the following line to archint.ini:

```
TRACEFILE <file name>
```

To control the maximum size of the trace file, use the following setting in the archint.ini file:

```
TRACEMAX <size in KB>
```

Figure 9-4 shows the sample content of a archint.trace file.

```
c14:51:27 08/29 [00000] archpro.c          [ 9329]: TraceOpenSockets(): socket 424
is contained in [FDS_ACTIVE, FDS_UNICODE_PORT], connected to 'CM agent',
uid='CM-AGENT_1'
14:51:27 08/29 [00000] archpro.c          [ 9329]: TraceOpenSockets(): socket 432
is contained in [FDS_ACTIVE, FDS_UNICODE_PORT], connected to 'worker thread',
uid='CM-AGENT_1_w0'
14:51:27 08/29 [00000] archpro.c          [ 9329]: TraceOpenSockets(): socket 440
is contained in [FDS_ACTIVE, FDS_UNICODE_PORT], connected to 'worker thread',
uid='CM-AGENT_1_w1'
14:51:27 08/29 [00000] archpro.c          [ 9329]: TraceOpenSockets(): socket 448
is contained in [FDS_UNICODE_PORT]
14:51:27 08/29 [00000] archpro.c          [ 9215]: CloseAllSocketsToProcess():
changing rc=CS_RC_CLOSE_SOCKET to rc=CS_RC_OK
14:51:27 08/29 [00000] archpro.c          [ 9221]: CloseAllSocketsToProcess():
leaving with rc = CS_RC_OK (0)
14:51:27 08/29 [00000] archpro.c          [ 5293]: Archpro:
service_unicode_socket() (sock=416) returns failure CS_RC_SOCKET_PROBLEM (-10000):
```

Figure 9-4 Sample archint.trace file content

9.3.2 CommonStore server startup trace file (startup.trace)

CommonStore server startup trace file, startup.trace, contains trace entries, which are written during the startup of a CommonStore server. When startup trace is turned on, all CommonStore server executable record messages during the initial startup phase in this file.

In combination with REPORT ON (for console output), it is useful to find problems that occur during the server's startup phase.

To specify the startup trace file, add the following line to the archint.ini file:

```
STARTUP_TRACEFILE <file name>
```

Example for CSX:

```
STARTUP_TRACEFILE C:\CSX\startup.trace
```

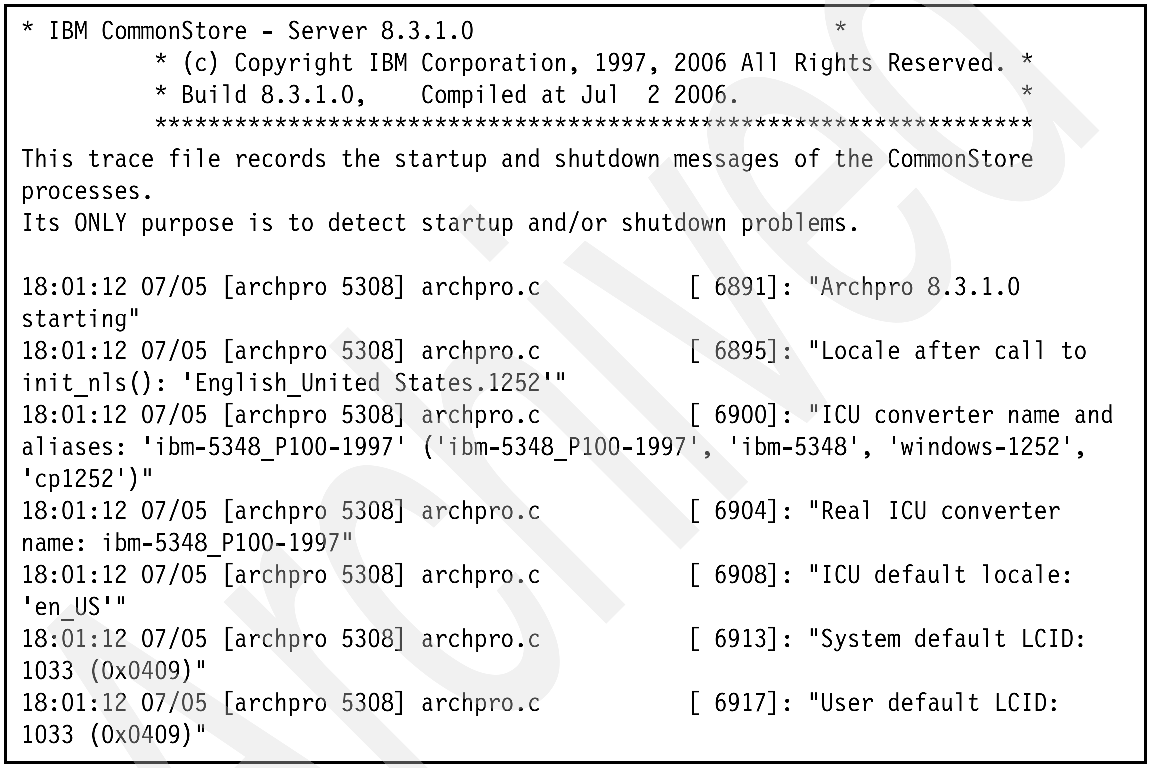
Example for CSLD:

```
STARTUP_TRACEFILE C:\CSLD\startup.trace
```

If it is not used, the http task tries to write the httpstartup.trc to a directory where archpro.exe is started or to the %TMP% directory.

> **Note:** The startup trace file contains all the tracing information created by the archpro during the startup. It is always created. The creation starts even before any configuration data is read. The startup trace file grows with each start of the CommonStore Server. Monitor its size and clean it up regularly. You can limit the size of the trace file by setting the traceMax value in the archinit.ini.

Figure 9-5 shows the sample content of a startup.trace file.



```
* IBM CommonStore - Server 8.3.1.0                          *
          * (c) Copyright IBM Corporation, 1997, 2006 All Rights Reserved. *
          * Build 8.3.1.0,    Compiled at Jul  2 2006.                     *
          **************************************************************
This trace file records the startup and shutdown messages of the CommonStore
processes.
Its ONLY purpose is to detect startup and/or shutdown problems.

18:01:12 07/05 [archpro 5308] archpro.c           [ 6891]: "Archpro 8.3.1.0
starting"
18:01:12 07/05 [archpro 5308] archpro.c           [ 6895]: "Locale after call to
init_nls(): 'English_United States.1252'"
18:01:12 07/05 [archpro 5308] archpro.c           [ 6900]: "ICU converter name and
aliases: 'ibm-5348_P100-1997' ('ibm-5348_P100-1997', 'ibm-5348', 'windows-1252',
'cp1252')"
18:01:12 07/05 [archpro 5308] archpro.c           [ 6904]: "Real ICU converter
name: ibm-5348_P100-1997"
18:01:12 07/05 [archpro 5308] archpro.c           [ 6908]: "ICU default locale:
'en_US'"
18:01:12 07/05 [archpro 5308] archpro.c           [ 6913]: "System default LCID:
1033 (0x0409)"
18:01:12 07/05 [archpro 5308] archpro.c           [ 6917]: "User default LCID:
1033 (0x0409)"
```

Figure 9-5 Sample startup.trace file

9.3.3 Content Manager Version 8 agent trace file (cmtrace.trc)

Content Manager version 8 agent trace file, cmtrace.trc, contains the tracing information for Content Manager Version 8 agent operation. The file is in a directory specified by the <*INSTANCEPATH*> parameter. Even if the TRACE OFF is requested, the Content Manager Version 8 agent writes startup information to the cmtrace.trc file.

The cmtrace.trc file includes the following tracing information:

- Timestamp
- Process ID
- Module name
- Trace message

Figure 9-6 shows the sample content of a cmtrace.trc file.

```
[22:23:26.740 22-08]  Entering CSConnectionICM.viewName(String)
[com.ibm.esd.commonstore.repository.icm.CSConnectionICM]
[22:23:26.750 22-08]  Leaving CSConnectionICM.viewName(String) (duration: 10 ms)
[com.ibm.esd.commonstore.repository.icm.CSConnectionICM]
[22:23:26.750 22-08]  Entering CSConnectionICM.viewName(String)   ...

[22:23:26.760 22-08]  CM-AGENT_1: ****** entering method 'sender.initialize'
[com.ibm.esd.commonstore.task.CSSocketMsgSender]
[22:23:26.760 22-08]  CM-AGENT_1: ****** entering method 'sender.logonToServer'
[com.ibm.esd.commonstore.task.CSSocketMsgSender]
[22:23:26.780 22-08]  CM-AGENT_1: logging on to CommonStore Server (host:
localhost, port: 3431): o.k.   [com.ibm.esd.commonstore.task.CSSocketMsgSender]
```

Figure 9-6 Sample cmtrace.trc file content

9.4 CSLD task and crawler logs

In this section, we cover the CommonStore for Lotus Domino task and crawler logs.

9.4.1 CSLD task log

CSLD task is a program that directly interacts with the Lotus Domino environment. It looks for jobs in the CSLD job database, which is the place where all user requests are collected before they are processed further. The CSLD task converts the jobs or requests, which are Lotus Notes documents, into files and passes the files to the Domino dispatcher. You can run several instances of the CSLD task at the same time.

The *CSLD task log files* are always written. Restarting the CSLD task overwrites the existing files. The file size is limited to a maximum of 25 MB. When this limit is reached, the first 15 MB of the file is removed.

The naming convention for the CSLD task log is cs<*yyyymmdd*>.log, where:

- *yyyy* indicates the year of the log.
- *mm* indicates the month of the log.
- *dd* indicates the day of the month of the log.

For example: cs20060808.log

Use the following command to start a CSLD task instance:

```
csld -n <config db> -s <domino server> -p <profile>
```

Each instance of the task writes an operation log file every day. The log files are created in the log directory. Specify the log directory in the task administration data object related to a task instance in the CSLD. The default log directory is C:\Program Files\IBM\CSLD\TASK\log. If you have more than one task instance, specify a different log directory for each instance. This avoids log files with the same name overwriting each other.

Use the startup log file for the verification of the settings that the task starts with. If this file is not written, use startup trace to check previous errors.

9.4.2 CSLD crawler log

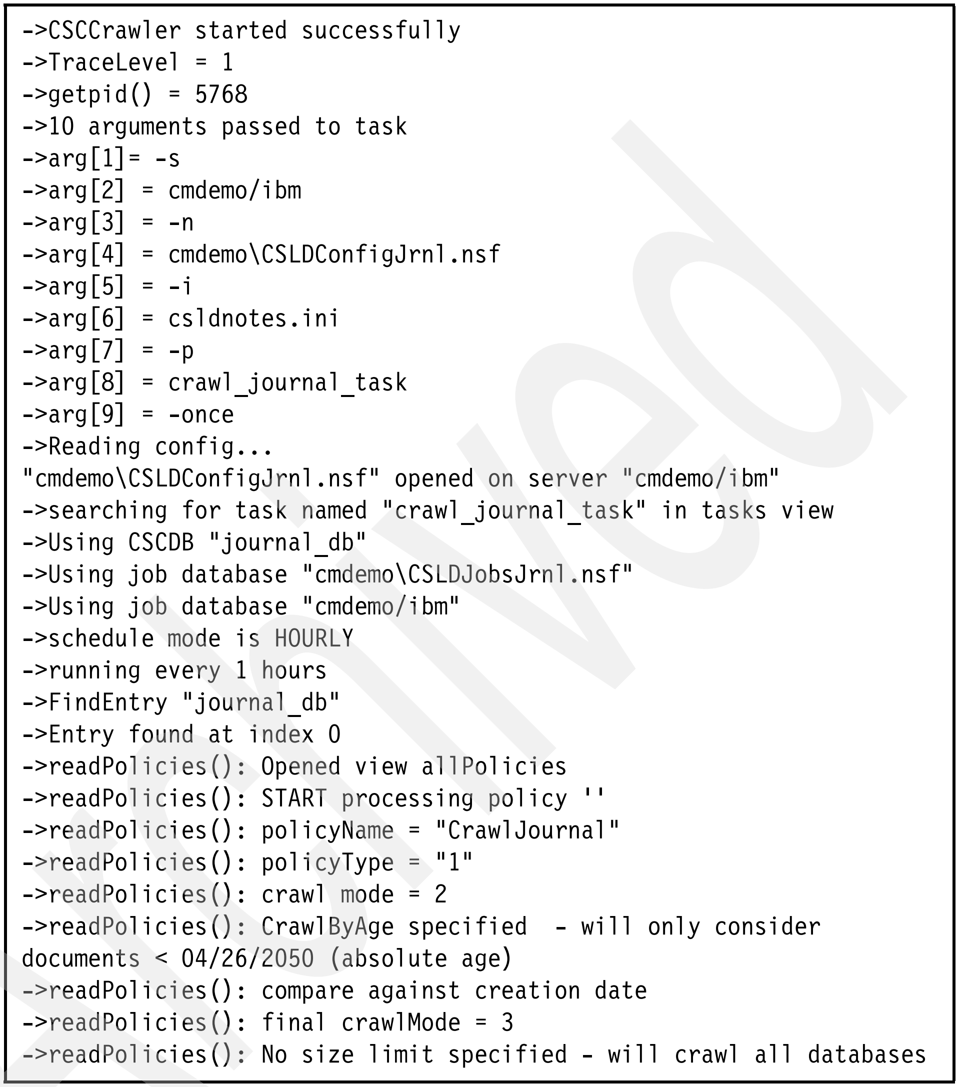
CSLD crawler is a program that is responsible for the creation of jobs in connection with the automatic functions of CommonStore. The crawler directly accesses the databases on your Lotus Domino servers and looks for documents that match the criteria specified in your policies. It then creates corresponding archiving and deletion jobs, and also retrieval jobs that are centrally triggered by an administrator. The crawler just creates jobs. The processing of the jobs is performed by CSLD task instances.

Use the following command to start a CSLD crawler instance:

```
csc -n <config db> -s <domino server> -p <scheduled task> -once
```

If everything is properly configured, the CSLD crawler immediately starts selecting documents greater than 1 KB from the mail database of the selected user. In a subsequent step, archiving jobs are created for these documents. The -once parameter ensures that the crawler instance runs just one time rather than every hour (as configured in the scheduled task document).

Figure 9-7 shows a sample CSLD crawler log.



```
->CSCCrawler started successfully
->TraceLevel = 1
->getpid() = 5768
->10 arguments passed to task
->arg[1]= -s
->arg[2] = cmdemo/ibm
->arg[3] = -n
->arg[4] = cmdemo\CSLDConfigJrnl.nsf
->arg[5] = -i
->arg[6] = csldnotes.ini
->arg[7] = -p
->arg[8] = crawl_journal_task
->arg[9] = -once
->Reading config...
"cmdemo\CSLDConfigJrnl.nsf" opened on server "cmdemo/ibm"
->searching for task named "crawl_journal_task" in tasks view
->Using CSCDB "journal_db"
->Using job database "cmdemo\CSLDJobsJrnl.nsf"
->Using job database "cmdemo/ibm"
->schedule mode is HOURLY
->running every 1 hours
->FindEntry "journal_db"
->Entry found at index 0
->readPolicies(): Opened view allPolicies
->readPolicies(): START processing policy ''
->readPolicies(): policyName = "CrawlJournal"
->readPolicies(): policyType = "1"
->readPolicies(): crawl mode = 2
->readPolicies(): CrawlByAge specified  - will only consider
documents < 04/26/2050 (absolute age)
->readPolicies(): compare against creation date
->readPolicies(): final crawlMode = 3
->readPolicies(): No size limit specified - will crawl all databases
```

Figure 9-7 Sample CSLD crawler log

Note: CSLD task and crawler also have trace. The trace can be enabled in the configuration database. In this chapter, CSLD task and crawler's trace are not discussed.

9.5 CSX logs and traces

In this section, we cover the CommonStore for Exchange Server logs and traces.

9.5.1 Controlling CSX task

Although this section focuses on the logs and traces, understanding how to control CSX task is important in determining what information will be logged and what activities will be traced.

You can control and tune the behavior of the CSX task through settings in the task's properties file. For example, if you do not archive from public folders, you can exclude public folders from crawling. The task's properties file does not come by default installation. You can create and store it in the <*installation_path*>\Task directory. The file name is <*task_name*>.properties.

You can set the following parameters in the task's properties file:

- CSX_TASK_DISABLE_POLLING

  If set to YES, no polling thread is started for the task.

- CSX_TASK_DISABLE_CRAWLING

  If set to YES, no crawler thread is started for the task.

- CSX_TASK_DISABLE_CRAWLING_MAILBOXES

  If set to YES, crawling is only done for public folders.

- CSX_TASK_DISABLE_CRAWLING_PUBLIC_FOLDERS

  If set to YES, crawling is only done for mailboxes.

- CSX_TASK_CRAWLING_CYCLES=n

  This specifies how often the crawler crawls the list of mailboxes and public folders:

  – n=0: Lists the mailboxes and public folders to be stored; when n is set to 0, there is no crawler activity performed on the mailboxes or public folders.

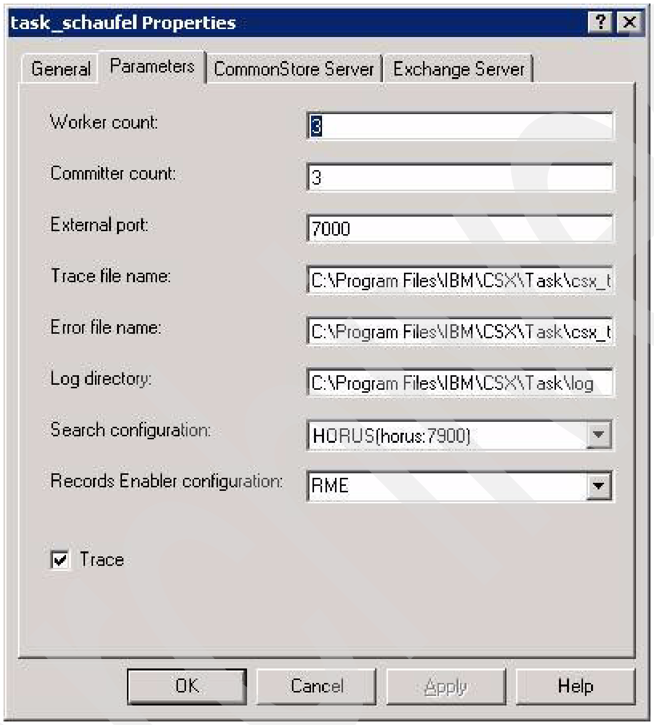
  – n=1: Performs exactly one crawling cycle; this helps to create reproducible scenario and to reduce the chance of log wrapping when debugging problems.

Note: Set the variables listed previously only when debugging the CommonStore solution.

You can configure the following settings in CSX System Manager graphical user interface (GUI):

- CSX task log directory
- CSX trace file name
- CSX error file name

Figure 9-8 shows the Parameters tab of a task property in the CSX System Manager.



Worker count: 3

Committer count: 3

External port: 7000

Trace file name: C:\Program Files\IBM\CSX\Task\csx_t

Error file name: C:\Program Files\IBM\CSX\Task\csx_t

Log directory: C:\Program Files\IBM\CSX\Task\log

Search configuration: HORUS(horus:7900)

Records Enabler configuration: RME

☑ Trace

Figure 9-8 CSX System Manager

To turn on the traces, select the **Trace** check box from this tab. When you are done with tracing, make sure that you disable tracing and delete the trace files afterwards.

Controlling CSX crawler activities

You can set variables to control the crawler activities. These variables are set in the Windows environment variable panel.

The following variables limit crawling to individual stores or sets of stores:

```
CSXCRAWLERINCLUDE
…MAILBOXALIAS
…MAILBOXALIASPREFIX
…PUBLICFOLDER
…PUBLICFOLDERPREFIX
```

For example, when you set `CSXCRAWLERINCLUDEMAILBOXALIAS=TESTUSER`, it forces the crawler to work on the mailbox of the user with the alias name TESTUSER only. All other mailboxes are ignored.

You can exclude individual stores or sets of stores from being crawled using the variables:

```
CSXCRAWLEREXCLUDE
…MAILBOXALIAS
…MAILBOXALIASPREFIX
…PUBLICFOLDER
…PUBLICFOLDERPREFIX
```

For example, when you set `CSXCRAWLEREXCLUDEMAILBOXALIASPREFIX=TEST`, it forces the crawler to ignore the mailboxes of users where the alias name starts with TEST even if they have a policy assigned.

CSXCRAWLERNOJOBS controls generating jobs. If it is set to any value, no jobs are generated, but crawling and logging are still done. For example, when you set `CSXCRAWLERNOJOBS=1`, no jobs are generated. These include any archiving, restubbing, and declaring jobs.

9.5.2 CSX logs

The CSX task is the program that directly interacts with the Microsoft Exchange environment. It is responsible for performing interactive and automatic archiving and retrieval. It can be logically split into crawler, poller, worker, and committer components. We cover CSX task log and CSX crawler log files in this section.

CSX task log

The *CSX task log files* are always written. The naming convention for the CSX task log is cs<*yyyymmdd*>.log, where:

- *yyyy* indicates the year of the log.
- *mm* indicates the month of the log.
- *dd* indicates the day of the month of the log.

For example: cs20060807.log

Use the following command to start a CSX task instance:

```
csx <Task Name>
```

Each instance of the task writes an operation log file every day. The log files are created in the log directory. The default log directory is C:\Program Files\IBM\CSLD\TASK\log. If you have more than one task instance, specify a different log directory for each instance. This avoids log files with the same name overwriting each other.

Use the startup log file for the verification of the settings that the task starts with. If this file is not written, use startup trace to check previous errors.

CSX crawler log

The *CSX crawler log* file is a text file that contains information about the crawler's activities. Each activity is logged in an individual record of a specific record type. The file size is limited to a maximum of 25 MB (approximately 100,000 records). When this limit is reached, the first 15 MB of the file is removed.

Each crawler instance writes its own log file. It contains a list of the message stores (mailboxes or public folders) that it goes through. In addition, an entry for each message store is written to the log file. Each entry contains the name of the archiving policy that is assigned to the store and information about the crawler schedules. Each logging entry specifies if the message store was entered by the crawler.

CSX crawler log has the following file name format:

<*Log Directory*>\crawler_<*Task Name*>_(<*Exchange Server*>)_<*Suffix*>.log

Here <*suffix*> can be Mailboxes and PublicFolders and Journal_<Alias of journal recipient mailbox> if journaling is activated.

For example:

- C:\Prog...\IBM\CSX\TASK\log\crawler_mytask_server1_Mailboxes.log
- C:\Prog...\IBM\CSX\TASK\log\crawler_mytask_server1_PublicFolders.log

The crawler log files are split according to working areas. When crawling a mailbox, trace is written to a different file from the one used when public folders are crawled.

9.5.3 CSX traces

CSX task trace is similar to the tracing for CommonStore server. If something goes wrong, you can turn on tracing for defect analysis. However, tracing costs system performance and disk space. Crawler logging is always done and many errors can be found by analyzing the crawler log.

If an error occurs, try to reproduce the error with tracing switched on. If trace is on, task restart overwrites the existing files. The file size is limited to a maximum of 25 MB. As soon as this limit is reached, the first 15 MB of the file is removed. The next traces are appended. If you need to send trace file to the CommonStore support team, turn the trace on, restart the task, reproduce the error, stop the task, and send the collected trace file.

Note that if the task is started, it overwrites an existing trace file. This makes it difficult to catch errors. If the task is started as a service and encounters a crash, because the service controller realizes that the task is no longer running and restarts it, the restarting action overwrites the trace file. Therefore, we recommend that you do *not* start the task as a service for error analysis.

The trace files include:

- Crawler trace files

  <*Trace Directory*>\crawler_<*Task Name*>_(<*Exchange Server*>)_<*Suffix*>.trc

  Here <*Trace Directory*> is the directory of the file specified in the System Manager panel Trace file name. <*suffix*> can be Mailboxes and PublicFolders and Journal_<Alias of journal recipient mailbox> if journaling is activated.

  These crawler trace files are split according to working areas. When crawling a mailbox, trace is written to a different file from the one used when public folders are crawled.

  For example, the task starts multiple crawler threads for each server:

  – 1 thread that crawls all mailboxes. For example: crawler_mytask_server1_Mailboxes.trc

  – 1 thread that crawls all public folders. For example: crawler_mytask_server1_PublicFolders.trc

  – 1 thread for each journal recipient mailbox, because this mailbox has the highest activity. For example: crawler_mytask_server1_myJournaledMailbox.trc

- Polling trace

  <*Trace Directory*>\polling_<*Task Name*>_(<*Job Folder Name*>).trc

For example:

C:\Prog...\IBM\CSX\TASK\log\polling_mytask_server1_myJobFolder.trc

Traces for interactive archiving and retrieval. This trace is for tracking calling sequence leading to the internal task job creation.

- Job folder trace

  <*Trace Directory*>\job-folder_<*Task Name*>_(<*Job Folder Name*>).trc

  Access to job folder using JobFolder class. This trace focuses on synchronization of calls opening job messages and keeping track of job state.

  Job folder trace is for tracking down problems in opening the job folder for read/write access and changing single jobs.

- Operation trace of task instance, fully qualified file name is chosen in CSX System Manager

  Operation trace is for basic task processing (worker and committer).

- Startup trace

  Run csx.exe with the option `-t <fully qualified filename>`

  Startup trace records environment helps to locate possible problems when starting csx.exe.

- CSX Active Directory trace

  <*Log Directory*>\CsxADData_<*Task Name*>.trc

  This is a trace file holding Active Directory data as used by CSX task. Active Directory trace is for keeping track of the data (for example, rules), which is known to the task at a given point in time.

While the crawler trace needs to be checked for problems in automatic archiving, the polling trace and job folder trace show activities for interactive processing.

Additional traces include:

- CSX System Manager traces

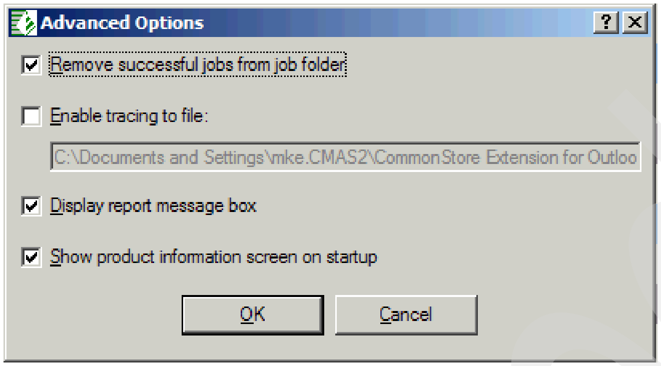
  The trace file name is CSX.TRC in the directory %USERPROFILE%\CommonStore for Exchange Server System Manager.

  Traces the activity of the CSX System Manager. To enable it, set the environment variable, `CSXMMCTRACEON` to `1`. If the CSX System Manager is active at the time that you set the variable, you must restart it.

- CSX Outlook extension trace

  The trace is always written to C:\documents and settings\currentUser\CommonStore Extension for Outlook.

To enable, select **Tools** → **Options** → **CommonStore** → **Advanced Options**. See Figure 9-9.


Advanced Options

☑ Remove successful jobs from job folder

□ Enable tracing to file:

C:\Documents and Settings\mke.CMAS2\CommonStore Extension for Outloo

☑ Display report message box

☑ Show product information screen on startup

Figure 9-9 Enabling CSX Outlook extension trace

- CSX pst file archiving trace

  Parameter when starting the tool, [-trace], turn on the trace.

  Trace files are in the subdirectory CommonStore CSXPST of the directory specified by environment variable `USERPROFILE`. Usually, this is your Documents and Settings directory. If this is not set, they are in the subdirectory of directory specified by TMP or TEMP. If none of these environment variables are set, a subdirectory is created on drive C:\.

- CSX Search Server traces

  These traces are via Web server and in a directory such as E:\Prog...\IBM\CSX\Search Server\logs\server1.

  The trace files are those that are usual for the Web server such as SystemOut.log and SystemErr.log.

- Search user exit
- Outlook Web Access user action traces

  These trace files include the Outlook Web Access ASP.NET page trace file and the Outlook Web Access Web Service trace file.

For more information about these traces, see the latest product manual.

9.6 Report logging feature (CSLD)

In this section, we discuss the CommonStore report logging feature. CommonStore for Lotus Domino offers a *report logging* feature that can record various CommonStore task operations to log files. You can import these files to other applications, such as Microsoft Excel®, and produce customized reports to analyze CSLD task processing or for other business requirements.

When report logging is enabled, CSLD task instances can log events such as archiving, retrieval, deletion, update, and search operations. The logging feature has minimal impact on the overall performance of the CSLD task. It also has an error-handling capability that ensures the task operations are logged.

Currently, there is no such feature supported for CommonStore for Exchange Server.

9.6.1 Report logging files

Every day the report logging feature generates a log file that contains the operations performed by all the CSLD tasks that are running on the same logical system. Multiple tasks from the same system write to the log file.

The naming convention for the report log is ld<*key*><*yyyymmdd*>.log, where:

- *key*: Uniquely identifies the log.
- *yyyy*: Indicates the year of the log.
- *mm*: Indicates the month of the log.
- *dd*: Indicates the day of the month of the log.

Example:

```
ldMYKEY20060808.log
ldMYKEY20060809.log
```

The <*key*> value can be any alphanumeric characters (including underscores). Multiple physical CSLD tasks that belong to the same logical CSLD system must use the same <*key*> value so that all the CSLD task operations of the same logical system are logged in one place. The <*key*> ensures that the report logging files include all task operations of a logical CSLD system. The <*key*> value is also used by the report logging daemon and the logging daemon control program as described later in the chapter.

The report logging files are plain text files in CSV format (comma separated values). Within the files, one line contains one record. Each record represents the log entry of a task operation. Each column (separated by comma) within each log entry contains one data item. The report log entry is defined in Table 9-2.

Table 9-2   Report log entry definition

Report data	Comments
Report Entry Identifier	Unique ID for the report log entry
Archive ID	Determined by CSLD task through mapping
Document ID	Returned by CommonStore server to the CSLD task after archiving
Operation	Data collected depends on the operation; operations include Archive, Delete, Update, Retrieve, and Search
Processing Start Time	This is the timestamp of when the CSLD task started processing this document (read its UNID for the job). The timestamp is in hh:mm:ss:nn format, where hh represents hour, mm represents minute, ss represents second, and nn represents millisecond.
Processing Duration	Duration timestamp (in millisecond) when this particular document is finished processing
CS Server Return Code	The primary return code from the CommonStore server
Originator	Job requester-CSLDOrgUser Note: This assumes that the jobs created by the crawler sets the CSLDOrigUser to the correct owner. The CSLD task just gets the CSLDOrigUser.
Database Server	Domino database server
Database Path	Database path
Archiving Type	It can have one of the following values: Attachment, Entire, Component, or Convert
Archiving Format	It can have one of the following values: Notes, DXL, or empty
Document Size Before Archiving	Size (in bytes) of the Notes document to be archived; this data size is the Notes data size If the archiving method type is Attachment, then this value is the attachment size. If the archiving type is Entire, then this value is the Notes document size. If the archiving type is Component, this is a bit tricky: The native component's size is the size of the Notes document after the attachments are detached. In all cases, this size can be obtained from the Notes APIs.
Total Data Size Archived/Retrieved	File sizes of the data sent to the archive; it is the file size of the individual components files (each part that is archived is stored to a file, hence the file size is the data size archived)
SIS CDI	This stands for Single Instance Store Computed Document Identifier. This is the same for *all* copies of one document.

Report data	Comments
SID CRI	This stands for Single Instance Store Computed Record Identifier. This is different for each copy of the same document.
Postprocessing	This contains one of the following string: ► remove embedded ► stub document ► delete original Remove embedded is the same as remove embedded and attachment
Document UNID	Notes document ID
Component ID	Depending on the archive definition and when component archiving is done, there is one document ID and several component IDs. Therefore, the external report program can group the information about the multiple components within one document ID.
Content Type	This is determined by mapping; if none is defined, the CSLD takes the file extension and the CommonStore server either takes an existing mapping or creates one.
Number of Components	This is the number of documents in this job. It can be multiple documents that are identified by the component ID per one document ID.
File Name	This is the source file name when logging an archive operation, and target file name on retrieve operation.

For each CSLD task operation (Archive, Retrieve, Delete, Domino search, and Delete), Table 9-3 shows what data is gathered in the report log files. The X marks under the tasks indicate that the data (on the left) is reported for the tasks. For example, the log file gathers Report Entry Identifier, Operation, Processing Start Time, and Processing Duration information for all operations. Because each report log entry has fixed data columns, some CSLD task operations may not generate any information for a particular data column. For example, the report data, Archiving Type in Table 9-3 (the first column) is only applicable to the Archive operation. For the Retrieve, Delete, Domino search, and Update operations, this data field will always be empty.

Table 9-3 Data gathered per CSLD task operation

Report data: Log entry	Archive	Retrieve	Delete	Domino search	Update
Report Entry Identifier	X	X	X	X	X
Archive ID	X	X	X	X	X
Document ID	X	X	X	X	X

Report data: Log entry	Archive	Retrieve	Delete	Domino search	Update
Operation	X	X	X	X	X
Processing Start Time	X	X	X	X	X
Processing Duration	X	X	X	X	X
CS Server Return Code	X	X	X	X	X
Originator	X	X	X	X	X
Database Server	X	X	X	X	X
Database Path	X	X	X	X	X
Archiving Type	X				
Archiving Format	X				
Document Size Before Archiving	X				
Total Data Size Archived/Retrieved	X	X		X	
SIS CDI	X				
SID CRI	X	X	X	X	X
Post processing	X		X		
Document UNID	X	X	X	X	X
Component ID	X			X	
Content Type	X	X		X	
Number of Components	X	X		X	
File Name	X	X		X	

Figure 9-10 shows a sample content of a report log file. This sample log has *five* lines. The first line is the header indicating the data columns to be reported in the log file. It occurs only once in the log file. The next four lines show the log entries of the four operations. In this sample, these operations are all Archive operations. The first operation started at 14:34:56.20, and it took 33 milliseconds to complete. The report ID for the operation is 44AD02EE4C1738EEOC. The server where the e-mail resides is centaurius/CSLDDev. The source of the e-mail is mail\ctest1.nsf database. The archiving type is Entire. Notice that some of the fields are empty and all columns are separated by commas. The last operation, with the report ID of 44AD0B960708A4EE0C, started at 15:10:08.16 and it took 68 milliseconds to complete.

```
REPORT_ID, ARCHIVE_ID, DOCUMENT_ID, OPERATION, START_TIME, DURATION[MS], CS_RC,
ORIGINATOR, SERVER, SOURCE, TYPE, FORMAT, DOC<ARC, ARCHIVED, CDI, CRI, POSTPROC,
DOC_UNID, COMP_ID, CONT_TYPE, #COMPS, FILENAME
44AD02EE4C1738EE0C,,,Archive,14:34:56.20,33,-1,CN=CSLD
Test1/O=CSLDDev,centaurius/CSLDDev,mail\ctest1.nsf,Entire,Notes,5256,0,,,stub
document,FCCED30DE8FCF4EAC125716500522752,,,,
44AD042B3310C0EE0C,,,Archive,14:39:30.12,23,-1,CN=CSLD
Test1/O=CSLDDev,centaurius/CSLDDev,mail\ctest1.nsf,Entire,Notes,5225,0,,,stub
document,FCCED30DE8FCF4EAC125716500522752,,,,
44AD0A10390360EE0C,,,Archive,15:03:56.59,53,-1,CN=CSLD
Test1/O=CSLDDev,centaurius/CSLDDev,mail\ctest1.nsf,Entire,Notes,5233,0,,,stub
document,FCCED30DE8FCF4EAC125716500522752,,,,
44AD0B960708A4EE0C,,,Archive,15:10:08.16,68,-1,CN=CSLD
Test1/O=CSLDDev,centaurius/CSLDDev,mail\ctest1.nsf,Entire,Notes,5241,0
```

Figure 9-10 Report log file sample

> **Note:** After you turn on the report logging, one report is generated every day. It is a good practice to regularly monitor the size of these files and the disk space. Remove any report log files that you no longer require or offload them to some other location.

9.6.2 Logging daemon

In order to prevent data loss in case the CSLD task runs into a critical error situation, a separate process is created by CSLD automatically which is responsible to maintain the logging data even in cases where CSLD can no longer work correctly.

This separate process is called the *report logging daemon*. It is created automatically by the first CSLD task instance of a logical CSLD system and it is normally stopped by the last CSLD task instance. Its only purpose is to maintain the report logger data buffer. It maintains this data in shared memory so the different CSLD tasks can add their report logging data to it efficiently.

In order to connect the logical CSLD tasks with the daemon, they must share the same report logging <*key*> as specified previously.

The logging daemon regularly monitors the shared memory space. When the buffer is full, the logging daemon writes the data from the shared memory to the report log file on the disk. The report file uses the <*key*> value as part of its log file name. The logging daemon then clears the shared memory space and the CSLD task starts to write its entries to the shared memory space again.

When the CSLD task shuts down, it also shuts down the logging daemon. Before shutting down, the logging daemon flushes out all the data from the shared memory to the report log file. If there are multiple tasks written to the same shared memory, then only the shutdown of the last CSLD task results in log daemon to finally write out the data and stop working.

Figure 9-11 shows the logging daemon working in the CSLD system.

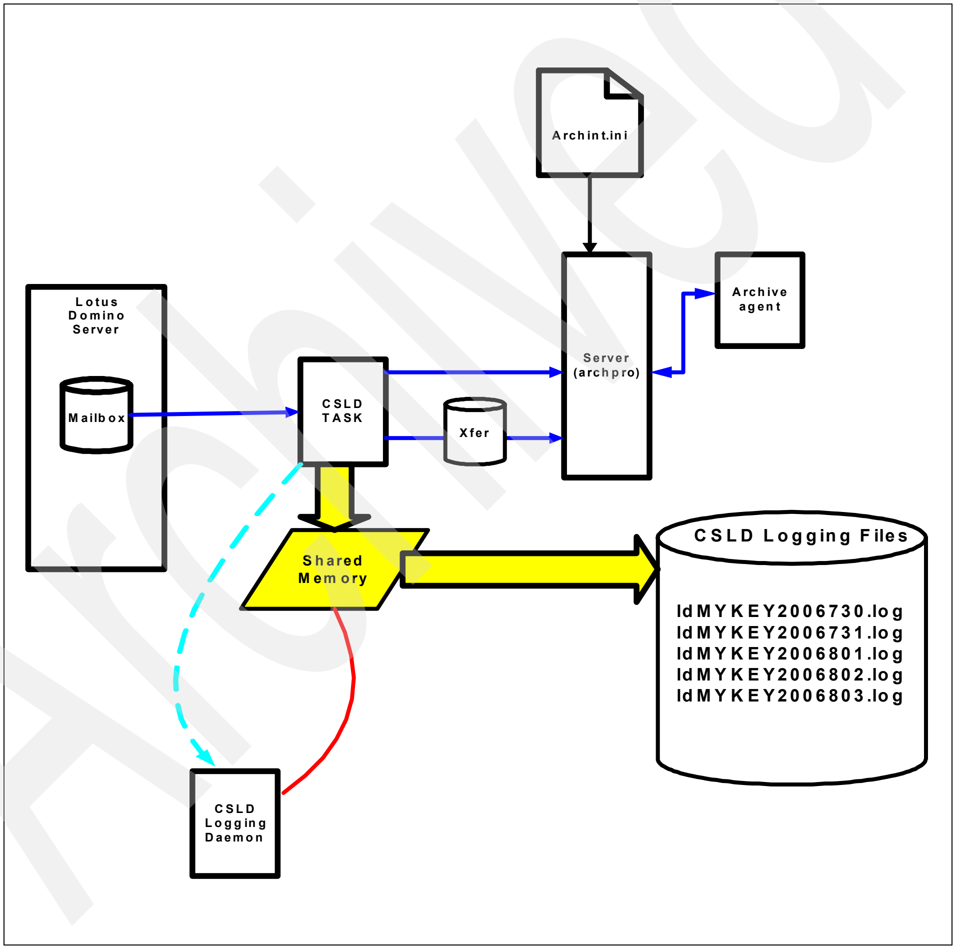


Figure 9-11 The logging daemon working within CSLD

The blue lines (single solid straight lines) in Figure 9-11 indicate the data flow of the messages to be archived. The yellow lines (thick lines with borders) indicate the report logging data flow. The red line (single curved line on the lower part of Figure 9-11) indicates that the logging daemon is holding and maintaining the shared memory into which the CSLD tasks write their logging data. The dotted blue line indicates that the CSLD task controls (in this example, starts and stops) the logging daemon.

The Xfer box in Figure 9-11 means the transfer directory with which the CSLD task and the CSLD server share data. This transfer directory must be configured in a CSLD system as described in the product manual.

9.6.3 Controlling CSLD report logging

The CSLD report logging feature uses several environment variables to control how it works. These variables control what operations to log, to what directory or folder to write the log file, and specify other user configurable data. Environment variables must be set in the **shell** command in which the CSLD task is launched.

The environment variables that control the report logging feature include:

- CSLD_LOGGING_KEY
- CSLD_LOGGING_ACTIVATE
- CSLD_LOGGING_REQUIRED
- CSLD_LOGGING_OPERATIONS
- CSLD_LOGGING_DIR
- CSLD_LOGGING_BUFSIZE

The CSLD_LOGGING_KEY variable defines the prefix of the log file. This is the <*key*> value that we mentioned previously. Its value can be a string of alphanumeric characters including underscores, for example, MY_KEY_123. Apart from its function as a file name prefix, the logging key is used to identify and reserve a portion of the memory (the shared memory space) for the logging operations of the task instances that use this prefix. In addition, each defined prefix causes the CSLD task to start a logging daemon in the reserved memory area.

The prefix is unique. If you set the same prefix for two or more task instances, these task instances will write log information to the same log file.

If the CSLD_LOGGING_KEY variable is not defined, report logging will be disabled.

Some of the considerations when setting up the CSLD_LOGGING_KEY variable are:

- Define a unique key for all CSLD task instances belonging to a logical CSLD system.
- Multiple CSLD task instances using the same CommonStore server *must* use the same key.
- One CSLD logging daemon is launched per logical CSLD system using this key.
- Its value can be a combination of alphabetic, digits, and underscore characters (for example, MYKEY).
- Invalid or absence of the key leads to error during the CSLD start.

> **Note:** If reporting is turned on in the archint.ini, but the variable CSLDLOGGING_KEY is not specified in the shell in which CSLD is launched, no report is generated.

The CSLD_LOGGING_ACTIVATE variable switches report logging `on` or `off`, depending on its value. A value of `yes` switches logging on. A value of `no` switches it off. If this variable is not present, the system defaults to `no`.

When setting its value, instead of using `yes`, you can also use the value of `1`, `on`, or `true`. When setting the value to `no`, you can also use `0`, `off`, or `false`.

The CSLD_LOGGING_REQUIRED variable specifies whether it is mandatory to log the CSLD task operations. A value of `yes` switches the option on.

If the option is set, this means that if the logging procedure fails for some reason, the CSLD task will not be allowed to do any more work. Any failure in logging operations will then result in:

- Notification of the system administrator (if they are set).
- Shutting down of the CSLD task.
- The logging daemon and its shared memory with logging records are still kept alive.

  The logging daemon must be shut down manually by the logging control program.

If the CSLD_LOGGING_REQUIRED is disabled, CSLD will continue even if a failure in the report logging occurs. As a consequence, some CSLD operations may not be logged and the log may be incomplete. For information about how the CSLD report logging feature handles mandatory logging and logging control program, see 9.6.4, “Error handling and the logging control program” on page 179.

When setting this value, instead of using `yes`, you can also use the value of `1`, `on`, or `true`. When setting this value to `no`, you can also use `0`, `off`, or `false`.

The CSLD_LOGGING_OPERATIONS variable sets the logging detail level based on the operation type. You can log information about one or more types of operations. You can set the operation type to any of the following values:

- archive
- delete
- update
- retrieve
- search
- all

To log more than one operation type but not all operation types, string up the appropriate values as you set the variable and separate the values by commas. For example, to log only archiving operations, use the following setting:

```
CSLD_LOGGING_OPERATIONS=archive
```

To log archiving, deletion, and updating operations, use the following setting:

```
CSLD_LOGGING_OPERATIONS=archive,delete,update
```

By default, all operations are logged. If you misspell the operation type, it will be ignored.

The CSLD_LOGGING_DIR variable sets the log file directory. You have to specify the fully qualified directory name, for example, D:\Program Files\IBM\CommonStore\CSLD\logging. The directory must exist and have write permissions upon the start of the CSLD task. If the variable is not set, the log files are written to the directory that the CSNINSTANCEPATH keyword points to in the archint.ini file. By default, this is the home directory of the CSLD user.

The CSLD_LOGGING_BUFSIZE variable sets the size of the memory buffer (the shared memory space) that is used for logging. The minimum size is 4 KB. The maximum size is 512 KB. The larger the buffer size, the more entries it can hold and less times for the data to be written to the log file. This reduces the performance impact of the writing process at the cost of reduced overall memory space available for other processes. Tune this setting based on your system configuration. If the variable is not set, or if the specified value is out of the allowed range, the buffer size will be 128 KB. To set the buffer size, specify values between 4 and 512.

To help you determines the right buffer size for the shared memory, use the this rule of guideline:

The average log entry is approximately 250 characters. Based on this assumption, 4 KB holds approximately 16 log entries, 64 KB holds approximately 260 entries, 128 KB hold approximately 520 entries, and 512 KB holds approximately 2100 log entries. Determine the size of the log buffer (shared memory) in KB based on how many log entries you want the shared memory to hold.

> **Note**: There is one report log file per day and the maximum log file size is not configurable.

Figure 9-12 and Figure 9-13 provide the sample batch file and script that you can use to set up report logging variables on Windows and AIX environments. You can run this before starting the CSLD task.

```
rem *******Begin Comment**************
rem This script will set the environment variables
rem for the rem CSLD reporting tool
rem *******End Comment**************
@echo off
set CSLD_LOGGING_ACTIVATE=3
set CSLD_LOGGING_REQUIRED=yes
set CSLD_LOGGING_OPERATIONS=update
```

Figure 9-12 Sample Windows script to set the report logging tool variables

```
#!/bin/sh
export CSLD_LOGGING_ACTIVATE=3
export CSLD_LOGGING_REQUIRED=yes
export CSLD_LOGGING_OPERATIONS=update
```

Figure 9-13 Sample AIX script to set the report logging tool variables

9.6.4 Error handling and the logging control program

The CSLD report logging tool provides provision for recovering information if an error occurs when the CSLD reporting logging daemon is active.

The possible reasons for report logging errors that occur are:

- Out of disk space on the logging directory
- The write permissions might be revoked on the logging directory or logging file
- The logging directory no longer exists
- The CSLD logging daemon cannot be launched

If CSLD logging daemon cannot be launched, check the permissions for the CSLD logging daemon executable (cslogdmn.exe). You can also try to restart the logging daemon manually to see if it is a valid executable file and the environment of the shell in which CSLD is launched has all the right setup. Probably, you might need to check your PATH and LIBPATH environment variables in the shell so that the CSLD logging daemon can be launched correctly.

Setting the CSLD_LOGGING_REQUIRED parameter to yes means that it is mandatory to log the CSLD task operations. When this is set and a report logging error occurs, the CSLD task shuts down automatically. The CSLD task does not shut down the logging daemon because the shared memory might still hold the logging entries that are not written to the report log file yet. You have to resolve the situation that caused the report logging error manually.

After you fix the problem, use *the logging control program* (cslogcontrol), a separate program, to manually force the logging daemon to write the data from the shared memory to the report log file and shut down the logging daemon. After this, you can restart the CSLD task manually, which reactivates the logging daemon.

This behavior is to ensure that when it is mandatory to log CSLD task operations, everything is logged to the report log or else no work gets done.

If the CSLD_LOGGING_REQUIRED parameter is set to no, then when there is a report logging error, the logging daemon is shut down automatically. CSLD continues processing. As a consequence, some CSLD operations may not be logged and the report log may be incomplete.

Figure 9-14 shows how the CSLD logging control program works within the CSLD system.

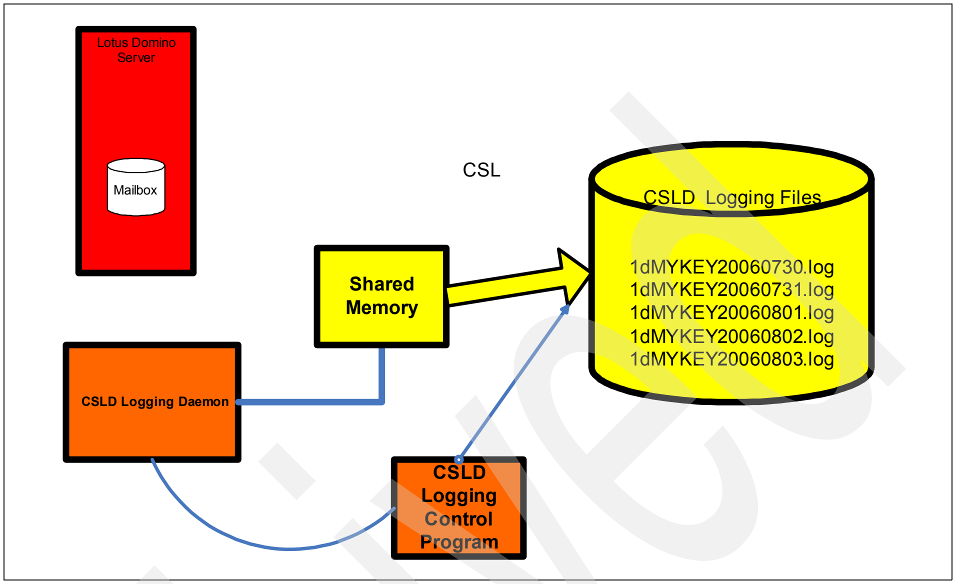


Figure 9-14 Error recovery within CSLD report logging

In Figure 9-14, the CSLD task is in a serious error condition and cannot control the report logging data nor the logging daemon any more. Luckily, the logging daemon maintains the shared memory in which still report logging data resides. This must be flushed (meaning written) to disk so that the report logging file accurately reflects all logging entries.

Since CSLD cannot trigger this anymore, the logging control program must be used manually to do so. The logging control program can be used to both write the contents of the report logging memory buffer to disk and then to shut down the logging daemon.

The logging control program syntax is:

```
cslogcontrol -key <key> {-dumptofile|-dstop|-dkill}
```

In this syntax:

- <*key*> is the key value for the shared memory space and it must be the value specified by the CSLD_LOGGING_KEY parameter.
- -dumptofile writes the data from the shared memory to the report log file.
- -dstop stops the logging daemon after all the CSLD tasks have shut down.
- -dkill immediately stops the logging daemon.

  Use this only as last resort when you cannot shut down the logging daemon gracefully (using -dstop). This option may not be called as long as CSLD tasks using the same key are still active; otherwise, you may get unexpected results.

For example, to force the logging daemon to write the remaining data from the shared memory to the log file, use the following command:

```
cslogcontrol -key MYKEY -dumptofile
```

After this is done, shut down the logging daemon:

```
cslogcontrol -key MYKEY -dstop
```

A

E-mail archiving questionnaires for solution planning and sizing

In this appendix, we include the standard e-mail archiving questionnaires for solution planning and sizing.

The questionnaires are:

- E-mail archiving questionnaires
  - General questionnaires
  - Mail server questionnaires
  - E-mail archiving for mailbox management questionnaires
  - E-mail archiving for compliance/discovery questionnaires
  - Supervision of e-mail content questionnaires
- eMail Search questionnaires
- Customer questionnaire for Content Manager
  - Content Manager component platform questionnaires
  - Content Manager system workload questionnaires
  - Client workload questionnaires
  - Content description and indexing questionnaires
  - Item processing and retrieval questionnaires

In addition to the questionnaires, we provide sample responses whenever applicable.

E-mail archiving questionnaires

Table A-1 lists a set of questionnaires related to e-mail archiving. For more information about these questions, see Chapter 3, “Solution design” on page 27, and Chapter 4, “Sizing” on page 45.

Table A-1 E-mail archiving questionnaire

Questions	Sample responses
Number of mail servers	8
Notes/Domino or Microsoft Exchange mail server?	Exchange
Number of users or mailboxes	22,000
Number of new e-mails per user per day	20
Percent of growth in average size of e-mails	10%
Percent of growth in the number of e-mail messages	10%
Will Single-Instance Storage (SIS) be used[a]?	yes
If yes, what is the average number of recipients an e-mail is sent within the company	2
Average size of the e-mail body or message text (KB)	100
Average size of the e-mail attachments (KB)	200
Average number of attachments	1
Percentage of e-mail with attachments	15%
How many old e-mails will be processed as part of an initial loading	1,000,000
What retention periods are required, and what percent of e-mails can be kept for each retention period?	
► 30 days	0%
► 60 days	0%
► 90 days	0%
► 1 year	0%

Questions	Sample responses
► 2 years	0%
► 3 years	0%
► 5 years	100%
► 7 years	0%
Number of daily hours available to complete e-mail capture	8
How many days can objects be kept on direct access storage device (DASD) managed by the Resource Manager before being migrated to Tivoli Storage Manager (external storage)?	90 days
Percent of mail messages to be full-text indexed[b]	100%
Percent of mail attachments to be full-text indexed (see Footnote b.)	0%
What retention managed storage device will be used?	IBM DR550
What percent of e-mails can be stored on retention managed storage versus other media?	100%
Will Content Manager replication be used?	No
Do you have an existing archive repository? If yes, please specify the archive software and platform. ► Content Manager (Windows, AIX, Solaris, IBM eServer iSeries™, or z/OS) ► Content Manager OnDemand (Windows, AIX, Solaris, iSeries, or z/OS) ► Tivoli Storage Manager (Windows, AIX, Solaris, or z/OS)	Content Manager (AIX)
If there is no existing archive repository, please indicate your archive repository and operating system preferences.	Content Manager (AIX)
Domino environment only Please indicate your CommonStore operating system preference.	CommonStore (AIX)

[a] Single-Instance Storage is supported only when Content Manager Enterprise Edition V8.3 for the following platforms, Windows, AIX, Solaris, or z/OS, is used for the archive repository.

[b] Full-text indexing and searching (keyword searches within attachments) is supported only when Content Manager Enterprise Edition V8.3 for the following platforms, Windows, AIX or Solaris, is used for the archive repository.

General questionnaires

Table A-2 lists a set of general questionnaires. For more information, see Chapter 3, “Solution design” on page 27, and Chapter 4, “Sizing” on page 45.

Table A-2 General questionnaires

Questions
What do you regard as your primary requirement for e-mail archiving? ► Managing the growth of the mail servers ► Compliance with corporate policies, industry regulations, or laws ► Compliance through detailed audit
When did you start using Lotus Notes/Domino or Microsoft Exchange?
Exchange environment only What is the status of your MS Active Directory implementation? ► Rollout completed ► Currently in test phase, rollout in the next 12 months ► No plans to move to MS Active Directory
Which other software besides the mail servers plays an important role in your company (for example, an enterprise resource planning (ERP) solution)? ► SAP® ► Siebel® ► Baan ► Movex ► Peoplesoft ► Other (please specify)
Domino environment only Do you also use Domino for applications other than messaging? ► Yes, some custom Domino applications ► Yes, some standard Domino applications (such as IBM Domino.Doc®) ► No
Domino environment only Do you already have an archiving system in place for other Domino applications? Are you interested in archiving documents or records from your Domino applications using the same CommonStore solution? ► Yes ► No
Exchange environment only Do you use Microsoft SharePoint® Portal Server? ► Yes, SharePoint Portal Server 2001 ► Yes, SharePoint Portal Server 2003 ► No

Questions
Do you already have an archiving system in place? ► Content Manager ► Content Manager OnDemand ► Tivoli Storage Manager ► Other (please specify)

Mail server questionnaires

Table A-3 lists a set of mail server related questionnaires. For more information, see Chapter 3, “Solution design” on page 27, and Chapter 4, “Sizing” on page 45.

Table A-3 Mail server questionnaire

Questions
Domino environment only What are the current release levels on the Domino server? Do you have any plans to move to a new release in the next 12 months? ► R5 ► R6 ► R6.5 ► R7
Exchange environment only What are the current release levels on the Exchange servers? Do you have any plans to move to a new release in the next 12 months? ► 5.5 ► 2000 ► 2003
How many mail servers do you run?
Do you run the mail servers clustered? ► Yes ► No
In how many sites are the mail servers located? Do you have any plans to consolidate the servers in the next 12 months?
Domino environment only Which operating systems are the Domino mail servers running on? ► Windows ► iSeries (OS/400) ► Linux ► AIX ► zOS ► Other (please specify)

Questions
Exchange environment only Which operating systems are the Exchange mail servers running on? ► Windows NT ► Windows 2000 ► Windows 2003
Which software do you use for backup of the mail servers? ► Tivoli Storage Manager ► EMC Legato Networker ► Veritas NetBackup ► Other (please specify)
Which anti-virus software do you use for scanning mail? ► Symantec ► Trend Micro ► Other (please specify)
What is the current disk capacity of all mail servers together? How much is already used by the mail databases?
How many GB per month is the current disk occupation growing on the mail servers?
Domino environment only Have you enabled full-text indexing for your mail databases on the Domino server? ► No ► Yes, but excluding file attachments ► Yes, including file attachments
Exchange environment only Have you enabled full-text indexing for your mailbox stores on the Exchange server? ► Yes ► No
Exchange environment only Do you use Exchange public folders? ► Yes, for document sharing and shared mailbox ► Yes, we have developed some custom forms and applications ► No

E-mail archiving for mailbox management questionnaires

Table A-4 lists a set of mailbox management related questionnaires. For more information, see Chapter 3, “Solution design” on page 27, and Chapter 4, “Sizing” on page 45.

Table A-4 E-mail archiving for mailbox management questionnaire

Questions
For how many of your users do you plan to introduce e-mail archiving for mailbox management? ► All ► Other (please specify)
How many of your users access their mail just in the office?
Domino environment only What is the primary access method for the office users? (Please also specify the release level in use.) ► Notes Client ► Webmail ► iNotes
Exchange environment only What is the primary access method for the office users? ► Outlook 2000 ► Outlook XP ► Outlook 2003 ► Outlook Web Access (OWA)
How many users are traveling and require remote mail access?
Domino environment only What is the primary access method for the traveling (mobile) users? (Please also specify the release level in use.) ► Local replica of their mailbox ► Webmail ► Domino Web Access (DWA)/iNotes
Exchange environment only What is the primary access method for the traveling (mobile) users? ► Outlook offline access (OST file) ► OWA
What is the average size of a user mailbox on the mail server? (in MB)
How much does the average mailbox grow per year? (in MB)

Questions
Domino environment only Has Notes database archiving been used for the mailboxes? (Please also specify an estimate of the total size of these databases.) ► No ► Yes, archive databases on the user PC ► Yes, archive databases on network drives ► Yes, archive databases on a secondary Domino server
Domino environment only Has your mail template been customized? ► Yes, by ourselves ► Yes, by Lotus services ► Yes, by IBM Business Partner ► No
Exchange environment only Has PST file archiving been used for the mailboxes? (Please also specify an estimate of the total size of these databases.) ► No ► Yes, PST files are on the user PC ► Yes, PST files are on network drives
How many years do you want to keep the e-mails in the archive system?
Do you charge the mailbox use to the departments? ► Yes (Please specify the monthly rate.) ► No
Have you set a quota on the user mailbox size? ► Yes (Please specify the quota in MB.) ► No

E-mail archiving for compliance/discovery questionnaires

Table A-5 lists a set of e-mail archiving questionnaires related to implementing long-term retention and discovery/compliance solutions. For more information about these questions, see Chapter 3, “Solution design” on page 27, and Chapter 4, “Sizing” on page 45.

Table A-5 E-mail archiving for compliance

Questions
For how many of your users do you plan to introduce e-mail archiving for compliance? ► All ► Other (please specify)
Which law, regulation, or policy do you have to comply with? ► Securities and Exchange Commission (SEC) ► NASD ► DoD 5015.2 ► HIPAA ► Sarbanes-Oxley Act ► FDA 21 CFR Part 11 ► Other (please specify)
Do you want to enable users to select specific e-mails and archive them on demand? ► Yes ► No
Do you want to enable users to categorize (adding search attributes) their e-mails as part of archiving? ► Yes ► No
Do you have to capture automatically all inbound and outbound (Internet) e-mails? ► Yes ► No
Do you also have to capture automatically all internal e-mails before being received by the user (mail journaling)? ► Yes ► No
Do you want to enable a central authority to investigate corporate e-mail communication (that is, doing a search in the mailboxes of several users)? ► Yes ► No
Do you also have to capture all instant messages (IBM Sametime® messages) ► Yes (please specify the instant messaging system)? ► No

Questions
When you subscribed to Bloomberg, do you have to capture all Bloomberg messages? ▸ Yes ▸ No
Should the user be able to categorize and archive not just e-mails, but also other business-related documents such as spreadsheets, Word documents? ▸ Yes ▸ No

Supervision of e-mail content questionnaires

The e-mail content supervision topic is beyond the scope of this IBM Redbook. Table A-6 lists the related standard questionnaires for your reference only.

Table A-6 Supervision of e-mail content questionnaire

Questions
For how many of your users do you plan to supervise the content of e-mails? ▸ All ▸ Other (please specify)
Do you want to stop the delivery of inbound and outbound e-mails based on the content of e-mail (real-time supervision, pre-review)? ▸ Yes ▸ No
Do you want to supervise the content of already sent mails based on a supervision policy (post-review)? ▸ Yes ▸ No
How many users will act as supervisors?
Do you have already defined rules for acceptable and unacceptable content for e-mails? ▸ Yes ▸ No
Do you want to log all supervisory activities? ▸ Yes ▸ No

eMail Search questionnaires

Table A-7 lists a set of questionnaires related to implementing eMail Search for CommonStore. For more information, see Chapter 3, “Solution design” on page 27 and Chapter 4, “Sizing” on page 45. For eMail Search, see Chapter 7, “eMail Search for CommonStore” on page 107.

Table A-7 eMail Search questionnaire

Questions	Sample responses
Number of mail servers	
Notes or Exchange?	Notes
Number of users or mailboxes	
Number of e-mails per user per day	
Total number of e-mails or messages per day	1,728,000[a]
Percent of growth in average size of e-mails	0%
Percent of growth in number of e-mails or messages	0%
Average number of recipients plus sender within the same mail domain	1
Average size of e-mail body or message text (thousand bytes)	5.0
Average size of e-mail attachments (thousand bytes)	60
Average number of attachments	1
Percentage of e-mail with attachments	100%
How many old e-mails can be processed as part of an initial loading	0
What retention periods are required, and what percent of e-mail can be kept for each retention period?	
► 30 days	0%
► 60 days	0%
► 90 days	0%
► 1 year	0%
► 2 years	0%
► 3 years	0%

Questions	Sample responses
► 5 years	100%
► 7 years	0%
Number of hours to complete capture	12
How many days will objects be kept on DASD managed by the Resource Manager before being migrated to Tivoli Storage Manager (external storage)?	0
What retention managed storage device will be used?	Centera
What percent of e-mail can be stored on retention managed storage versus other media?	100%
Will Content Manager replication be used?	No

a. Based on the calculation of 40 e-mails per second for 12 hours

Customer questionnaire for Content Manager

IBM Content Manager can be used as the back-end repository for CommonStore solutions. As the primary backbone of IBM Enterprise Content Management portfolio, Content Manager brings advanced content management functionality to the solutions. With features such as advanced data models, text search, hierarchical folder structures, workflow, open application programming interface (API), and document management capabilities, the repository excels at adding structure to traditionally unstructured content (including mail data archived using CommonStore).

Content Manager component platform questionnaires

Table A-8 lists a set of questionnaires related to Content Manager. Note that when sizing for eMail Search for CommonStore, Content Manager has already been taken into consideration.

The information provided here is for reference only.

Table A-8 Target platforms for Content Manager

Content Manager components	Responses	Comments
Library Server (Windows, Linux, AIX, or z/OS)	Win	Library Server is the central source for indexing, retrieving, managing, and organizing content.
Resource Manager (Windows, Linux, AIX, or z/OS)	Win	Resource Manager stores documents, images, and other resources.
Mid-tier server (Windows, Linux, or AIX)	Win	The mid-tier server manages connections to the Library Server and to the Resource Managers.

Content Manager system workload questionnaires

To estimate yearly volumes, information about daily or hourly workload is necessary for calculation. Table A-9 lists a set of system information questions that has to be answered.

Table A-9 Content Manager system workload questionnaires

Questions	Responses	Comments
How many hours daily will you be loading documents into the system?	10	
How many daily ad-hoc searches will your users be performing?	100	Enter the number of times that you expect the system will have to be searched for a document, folder, item, or resource to be retrieved.
How many business days per year will the system be available?	300	Typically 260
How many years do you want to size the system for?	5	

Client workload questionnaires

The stress on the system is calculated by multiplying the number of the clients by the number of actions they perform during the peak workload hour. By answering the questions listed in Table A-10, you can calculate the number of searches, retrievals, and processing actions that have to be performed by the mid-tier, Library Server, and Resource Manager during peak hour to support your business.

Keep in mind that after each action, a user might want to review the results (for example, look at the search result list or the retrieved document). As a general rule, a user doing more than thirty actions an hour (one every two minutes) is considered a heavy user.

Table A-10 Client workload questionnaires

Questions	Responses	Comments
How many Web clients and local area network (LAN) clients do you expect to be logged on during your heaviest use peak hour?	10	
How many times during the peak hour do you expect a client to: ► Conduct a Search ► Retrieve a document ► Update a document or folder (re-index or modify) ► Open a folder ► Add an item to a folder ► Start or advance a document or folder through a process	10 10 9 10	If you want to pass on workflow your data

Calculating the number of peak hours required

Note that the information included here is for reference only.

The system workload is calculated in two ways. The first way is from the user perspective by asking for how many users (clients), and how many actions each user performs during the peak hour. The second way is from a daily document processing perspective by asking how many documents are opened and how often they are opened. By comparing these two calculations, we can calculate an estimate of the number of daily peak hours it should take your users to process a daily workload.

```
Peak hours required = Daily Actions (such as opens, routes, and
searches) divided by Number of actions performed by the user
```

This number should be less than the number of daily hours that you expect to have the system available.

The calculated peak hours required can also show an error in your assumptions. For instance, if you have 1,000 users performing 10 document retrievals each hour, this translates to the system performing 10,000 document retrievals an hour. If you also have 80,000 new documents being retrieved three times each for processing, this translates to a system requiring to retrieve 240,000 documents each day. A simple comparison shows that your 1,000 users have to work 24 hours each day. You have to revisit your assumptions, and get this calculation down to something that is reasonable. In this case, if the system can be available for ten hours a day, and you think eight peak hours is a good estimate, you have to increase the number of clients to 3,000 or estimate that a document can be processed with just one retrieve.

Item description questionnaires

Note that the information included here is for reference only.

Items (such as documents, folders, items, and resource items) are assigned an item type when they are stored in the system. From the assigned item type, an item has user attributes and its content migration policy defined with it.
Table A-11 includes a set of questions that is used to describe your five largest input volumes of item types.

Table A-11 Item description questionnaires for large input volumes of item types

Questions	Responses	Comments
What is a descriptive name for the item type?	Document	Enter an identifying name for this item type.
What is the item's classification? (document, folder, item, or resource item)	doc	
How many items of this type can you create in the system each day?	1050	
What will be the total size of all the parts each item can have on average?	5000	
For documents and folders, how many other folders can each item be added to?	1	
For items and resource items, how many other items can each item be linked to?		
How many existing items will be loaded into the system when it is first installed?	100	

Content description and indexing questionnaires

Note that the information included here is for reference only.

Documents in the document model of Content Manager can have multiple parts. Typically, a document has a BASE part, sometimes an annotation part, and a separate note part. They can have multiple of each of these parts. If you store one hundred page document as a single TIFF file, the number of BASE parts is one. If you store the document as one hundred separate files, this number is one hundred. (BASETEXT parts are the same as BASE parts, but they can be index for full text search.)

Table A-12 includes a set of content description questions for sizing. You can enter a decimal fraction for the values too. If only half of your documents have annotations, the value of 0.5 is acceptable.

Table A-12 Content description

Content description, parts per item, versioning and storage	items	Comments
ICMBASE	1.0	Document only
ICMBASETEXT	1.0	Document only
ICMNOTELOG	1.0	Document or folder
ICMANNOTATION	1.0	Document only
ICMSTREAMVIDEO		Document only
How many versions of each part will you keep?	1	Should be at least one
How many days will the parts remain on DASD before migrating off to Tivoli Storage Manager?	90	
Will you be replicating the parts to a second Resource Manager?	No	

Table A-13 includes a set of indexing-related questions for sizing.

Table A-13 Indexing information

Questions	Responses	Comments
What will be the total length of all the attributes?	100	Sum of the length of the attributes in the item type
How many attributes will be designated as a database index?	2	Enter the number of key fields that will have a DB2 index placed on it. Typically one or two fields will have DB2 index.
What will be the total length of the attributes designated as database indexes?	20	Enter the sum of the length of one or two fields that will have the DB2 index on them.
What is the highest average number of values an attribute will have?	1.0	Example: If an item type has two attributes that can be multi-valued, one having an average number of three values, the other having an average value of five, the highest average number of values is five.

Item processing and retrieval questionnaires

The sizing tool calculates the number of daily opens as: ((Docs per days) multiplied by (new docs opened)) plus (daily ad-hoc opens). The sizing tool uses three numbers to calculate the daily number of item opens (folder opens or document retrieves) by the system:

- Number of new items per day
- Number of times each item is opened
- Number of items opened each day ad hoc

The sizing tool calculates the number of daily opens as: ((New items per day) multiplied by (new items opened)) plus (items opened ad hoc). Table A-14 includes the sizing questions to ask regarding item processing and retrievals.

Table A-14 Item processing and retrieval questionnaire

Questions	Responses	Comments
How many days will the items reside in the Content Manager DocRouting system?		How many items are on workflow process?
How many work nodes will each item be routed through?		Work node is one step in workflow.
How many times will each item be opened or retrieved for processing?		There are two ways of specifying item retrievals in the sizing tool. If you can make a statement such as: “Each item will be opened x-number of times during its processing”, you can enter that number x in this entry field. If you can make the statement as “Items of this type are retrieved x-number of times a day,” enter that number in the “daily ad-hoc opens” of this items. The first way is typical of Content Manager document processing. The second way is typical of ad-hoc opening from the users on environment such as help desk or after ad-hoc search. You can combine the two calculations as required.
How many items will be opened each day ad hoc?		

B

Additional material

This IBM Redbook refers to additional material that can be downloaded from the Internet as described below.

Locating the Web material

The Web material associated with this IBM Redbook is available in softcopy on the Internet from the IBM Redbooks Web server. Point your Web browser to:

```
ftp://www.redbooks.ibm.com/redbooks/SG247325addm
```

Alternatively, you can go to the IBM Redbooks Web site at:

```
ibm.com/redbooks
```

Select the **Additional materials** and open the directory that corresponds with the redbook form number, SG247325.

Using the Web material

The additional Web material that accompanies this IBM Redbook includes the following files:

File name	*Description*
SG247325.zip	Sample project outline and sizing exercise

System requirements for downloading the Web material

The following system configuration is recommended:

Hard disk space:	200 MB minimum
Operating System:	Windows
Processor:	Pentium® IV or later
Memory:	512 MB

How to use the Web material

Create a subdirectory (folder) on your workstation, and unzip the contents of the Web material zip file into this folder.

Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this IBM Redbook.

IBM Redbooks

For information about ordering these publications, see "How to get IBM Redbooks" on page 205. Note that some of the documents referenced here may be available in softcopy only.

- *E-mail Archiving and Records Management Integrated Solution Guide Using IBM CommonStore and Records Manager*, SG24-6795
- *Performance Tuning for Content Manager*, SG24-6949
- *Content Manager Implementation and Migration Cookbook*, SG24-7051
- *Content Manager Backup/Recovery and High Availability: Strategies, Options, and Procedures*, SG24-7063

Other publications

These publications are also relevant as further information sources:

- *Content Manager Enterprise Edition V8.3: Planning and Installing Your Content Management System*, GC27-1332
- *IBM WebSphere Application Server, Version 6.1 and IBM WebSphere Application Server Network Deployment, Version 6.1, data sheet*, G224-9140
- *Content Manager Enterprise Edition V8.3: System Administration Guide*, SC27-1335
- *CommonStore for Exchange Server V8.3.1 Administration & User's Guide*, SH12-6741
- *CommonStore for Lotus Domino V8.3.1 Administrators and Programmer's Guide*, SH12-6742
- *DB2 Net Search Extender Administration and User's Guide V9.1*, SH12-6842

Online resources

These Web sites are also relevant as further information sources:

- IBM CommonStore for Lotus Domino product information:

  http://www.ibm.com/software/data/commonstore/lotus/
- IBM CommonStore for Exchange Server product information:

  http://www.ibm.com/software/data/commonstore/exchange/
- Content Manager production information:

  http://www.ibm.com/software/data/cm/
- TechLine Sizing Support Portal (available to IBM representatives only):

  http://w3.ibm.com/support/americas/techline/sizewise.html
- Lotus documentation:

  http://www.lotus.com/ldd/notesua.nsf/find/domino
- DB2 production information:

  http://www.ibm.com/software/data/db2/
- WebSphere Application Server product information:

  http://www.ibm.com/software/webservers/
- WebSphere Application Server information center:

  http://www.ibm.com/software/webservers/appserv/infocenter.html
- Journaling with Exchange Server

  http://www.microsoft.com/technet/prodtechnol/exchange/guides/E2k3Journal/
- wikiCentral v2

  https://w3.webahead.ibm.com/w3ki/display/CSBP
- CommonStore download site - test fix

  ftp://ftp.software.ibm.com/software/commonstore

How to get IBM Redbooks

You can search for, view, or download IBM Redbooks, Redpapers, Hints and Tips, draft publications and Additional materials, as well as order hardcopy IBM Redbooks or CD-ROMs, at this Web site:

ibm.com/redbooks

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services

Index

IBM Redbooks

Best Practices for Setting Up an IBM CommonStore Solution

(0.2”spine)
0.17”<->0.473”
90<->249 pages

Best Practices for Setting Up an IBM CommonStore Solution

For Mailbox Management, e-Mail Retention and Discovery

Solution planning, design and sizing, and deployment

Journaling and eMail Search for CommonStore

Maintenance, logs, traces, and report logging feature

In today's business environment, e-mail has become one of the most commonly used forms of corporate communication. It is one of the core tools to conduct business transactions, and it generates a significant volume of critical business information including business discussions, decisions, and important business documentations. To manage the increasing volumes of e-mails, and to ensure long term e-mail retention, and provide discovery capability when required has become a real challenge that every organization faces today.

IBM CommonStore for Lotus Domino V8.3.1 and IBM CommonStore for Exchange Server V8.3.1 manage e-mail archiving and retrieval. With IBM eMail Search for CommonStore, they help to provide a solution that ensures e-mails are retained for a specified time and that they are full-text indexed and available to be searched when required.

This IBM Redbook provides guidance on setting up CommonStore solutions for mailbox management or long term e-mail retention and discovery purposes. We cover the topics including solution planning, design, sizing, deployment and configuration. We discuss journaling and eMail Search. We also discuss maintenance, troubleshooting, logs, traces, and the new report logging feature.

Although we do not have all the best practices in setting up CommonStore solutions, we provide some of the known practices and information that we collected at this point. This IBM Redbook serves as a starting point for you to explore ways to set up successful CommonStore solutions. Use this IBM Redbook in conjunction with the existing product manuals and solution documentation.

INTERNATIONAL TECHNICAL SUPPORT ORGANIZATION

BUILDING TECHNICAL INFORMATION BASED ON PRACTICAL EXPERIENCE

IBM Redbooks are developed by the IBM International Technical Support Organization. Experts from IBM, Customers and Partners from around the world create timely technical information based on realistic scenarios. Specific recommendations are provided to help you implement IT solutions more effectively in your environment.

For more information:
ibm.com/redbooks

SG24-7325-00 ISBN 0738494593