Inside the Lotus Discovery Server

Understanding the KDS architecture

Setting up a pilot

Deployment tips

ibm.com/redbooks
Inside the Lotus Discovery Server

December 2001
First Edition (December 2001)

This edition applies to versions 1.0 and 1.1 of the Lotus Discovery Server.

Comments may be addressed to:
IBM Corporation, International Technical Support Organization
Dept. TQH 1CP-5605E
1 Charles Park
Cambridge, Massachusetts 02142-1245

When you send information to IBM, you grant IBM a non-exclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

© Copyright International Business Machines Corporation 2001. All rights reserved.
Note to U.S Government Users – Documentation related to restricted rights – Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract with IBM Corp.
Chapter 7. Server recovery and troubleshooting .......................... 205
  7.1 Important - read the release notes first .......................... 206
  7.2 Why you need a server recovery plan ............................. 206
  7.3 Dealing with conflicts of interest ................................. 207
  7.4 Types of crashes .................................................. 209
    7.4.1 Degraded service ............................................. 209
    7.4.2 Partial failure ............................................... 210
    7.4.3 Complete failure ............................................. 210
  7.5 Using Windows tools to diagnose a problem ....................... 211
    7.5.1 Windows Task Manager ....................................... 211
    7.5.2 Windows Performance monitor ................................ 213
  7.6 Capturing crash information ...................................... 213
    7.6.1 Don’t destroy the evidence .................................. 214
    7.6.2 Screen dumps, errors and logs .............................. 214
    7.6.3 Recording incidents ......................................... 217
    7.6.4 Location of Lotus Discovery Server system files .......... 219
  7.7 Top ten configuration problems ................................... 221
    7.7.1 Common name = computer name = host name ................ 222
    7.7.2 Why don’t I have any affinities ........................... 225
    7.7.3 I can’t view/open my Lotus Discovery Server logs ......... 225
    7.7.4 Spidering a Web site never seems to complete ............ 226
    7.7.5 I can’t see any categories in my K-map .................... 227
    7.7.6 Cannot administer QuickPlace from Administration Place 228
    7.7.7 Server security settings are incorrectly set ......... 229
    7.7.8 KDSAdministrators group is missing important entries 229
    7.7.9 Access to the Domino Directory .............................. 230
    7.7.10 First-time start up error messages on LDS server console 231
  7.8 Looking for answers ................................................ 231
    7.8.1 Online documentation is context-sensitive .............. 232
    7.8.2 Compiled HTML help files ................................... 232
    7.8.3 Notes.net discussion forum .................................. 233
    7.8.4 Notes.net documentation library ............................ 233

Chapter 8. Planning a backup strategy for KDS .......................... 235
  8.1 Developing a backup strategy .................................... 236
    8.1.1 Backup architecture ......................................... 236
  8.2 Backing up distributed Discovery Services ..................... 239
  8.3 Backing up open files ............................................. 240
  8.4 Stopping and restarting KDS services ............................ 240
    8.4.1 Shutting down a Lotus Discovery Server .................. 241
  8.5 K-map - key backup points ....................................... 241
Preface

The Lotus Discovery Server is a knowledge server. It provides search and expertise location solutions designed to ensure that the relevant knowledge and collective experience of an organization is readily available to help individuals and teams solve everyday business problems.

To accomplish this, the Discovery Server extracts, analyzes, and categorizes structured and unstructured information to reveal the relationships between the content, people, topics, and user activity in an organization. It will automatically generate and maintain a Knowledge map (K-map) to display relevant content categories and their appropriate hierarchical mapping, which can easily be searched or browsed by users. The server also generates and maintains user profiles, and tracks relevant end-user activities, identifying those individuals who may be subject matter experts. Through this expertise profiling and content discovery, the server uncovers organizational know-how in terms of where things are, who knows what, what is relevant, and which subjects generate the most interest and interactivity.

In this IBM Redbook, we describe how to set up a successful Discovery Server implementation in an organization. We show you how to begin the project by looking at and sorting through the information you already have, how to install a pilot system you can use to learn how Discovery Server works with your information, and how to design your system to work effectively across multiple servers. We also touch on how this complex product works under the covers.

Attention: This book was originally written using Discovery Server 1.0. Since that time, Discovery Server 1.1 has begun shipping. The most notable difference between these versions is the absence of the K-Station portal product. While every attempt has been made to clarify the areas in the book related to K-Station, a few references to K-Station remain for those still using the initial version.

The team that wrote this redbook

This redbook was produced by a team of specialists from around the world working at the International Technical Support Organization, Cambridge Center.
David Morrison is an International Technical Support Specialist for Lotus Notes and Domino at the International Technical Support Organization Center at Lotus Development, Cambridge, Massachusetts. He manages projects and writes Redbooks on all areas of Lotus products. Before joining the ITSO in 1999, he was a senior Lotus Notes consultant working for IBM e-business services in the United Kingdom.

Peter Northam has been with the Australian Bureau of Statistics for 20 years and has been a key player in the introduction and implementation of Lotus Notes/Domino to the organization. As a senior systems administrator, Peter has also been responsible for the deployment of a number of Lotus products such as FxD, MSD and Domain Search, to provide innovative solutions and raise the operational effectiveness of the ABS. Over the last year, Peter has been involved in the development and implementation of various knowledge management initiatives as part of the ABS's commitment to excellence in knowledge sharing, innovation and high productivity. This includes evaluation of the Lotus Discovery Server since Test Build 3.

Martin Rueckert is an Lotus Professional Services consultant attached to the LPS-Central/Germany region. He has worked for Lotus Development for over four years on a part-time basis while studying Computer Science. He has a degree in Information Technology, writing his thesis on “Portal Technology and KM”. He has focused on Lotus KM products, particularly K-Station and the Lotus Discovery Server. He has worked with KDS since Test Build 3 at customer sites, supporting customers onsite in the KDS beta program and being a speaker at many Lotus/IBM KM sessions.

Lasisi Tabel is a Principal Certified Lotus Instructor for both administration and developer paths. During his three years at Lotus Development, attached to the LPS-Rhein/Main in Germany, he has focused on knowledge management-related products such as Sametime, QuickPlace, Domino.Doc and Domino.Workflow, as well as the Lotus Knowledge Discovery System. He has been involved in many KM enablement workshops and Master Classes on customer sites, as well as at Lotus/IBM, including the IBM Winter University 2001. He holds a degree in Business Administration and Computer Science and has 10 years of IT experience.

Thanks to the following people for their contributions to this project:

International Technical Support Organization, Cambridge Center

Glen Kelley, Lotus Development Corporation

David Kajmo, Lauren Wendel, Craig Wolpert and George Sprott

IBM Westford Lab
Special notice

This publication is intended to help a KDS deployment team to understand how the Lotus Discovery Server works in order to successfully install and deploy the product.

IBM trademarks

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

- e (logo)®
- IBM ®
- AIX
- AS/400
- AT
- CT
- DB2
- MQSeries
- Netfinity
- RS/6000
- S/390
- Shark
- SP
- SP2
- WebSphere

- Redbooks
- Redbooks Logo
- XT
- 400
- Lotus
- 1-2-3
- cc:Mail
- Lotus Notes
- Word Pro
- Domino
- iNotes
- Notes
- QuickPlace
- Sametime

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 Redbooks in one of the following ways:

- Use the online Contact us review redbook form found at: ibm.com/redbooks
- Send your comments in an Internet note to: redbook@us.ibm.com
- Mail your comments to the address on page ii.
Chapter 1. Introduction to the Lotus Discovery Server

The Lotus Discovery Server is the second product to come from the Lotus Knowledge Discovery System project, the first being K-Station. It is a back-end piece of knowledge management software designed to be used with portal software such as IBM WebSphere Portal Server.

This IBM Redbook provides an in-depth look at how the Lotus Discovery Server works. You will learn about how to set up a demonstration environment, how to plan for a full deployment of a Lotus Discovery Server environment, and how the internal components of the Lotus Discovery Server operate.

To get the most out of this book, readers should have an understanding of the discipline of knowledge management and what it means to their organization. You will also find the Lotus Discovery Server introduces new terminology, and you’re encouraged to familiarize yourself with the product by reading 1.1, “Essential terminology and definitions” on page 2.

In 1.2.2, “A guided question and answer tour” on page 6, we guide administrators through a tour of the Lotus Discovery Server by providing a question and answer discussion thread. This discussion, combined with knowledge of the new terminology, will help readers prepare for the following chapters which go into greater detail about how the Lotus Discovery Server works.
1.1 Essential terminology and definitions

The following terminology and definitions provide a brief set of “must know” terminology enablers of Lotus Knowledge Discovery System technology. This book makes extensive use of these terms, and your understanding of them will enhance your ability to interpret information written about the Lotus Discovery Server.

For a more detailed explanation of these terms, refer to the online documentation that is installed with the product

Knowledge management
At Lotus and IBM, we define knowledge management as a discipline to systematically leverage information and expertise to improve organizational responsiveness, innovation, competency, and efficiency.

Lotus Knowledge Discovery System
In its initial releases of 1.0 and 1.0a, Lotus Knowledge Discovery System shipped with the Lotus K-station portal server. However, since version 1.1, Lotus Discovery System is now a separate, self-contained product that can be integrated into any portal offering. Depending on your organization's needs, Lotus Discovery Server can be deployed as standalone products or as an integrated solution. K-station and the IBM WebSphere Portal Server are being merged to become the portal server to suit all business requirements.

The Lotus Discovery Server (or Discovery Server) is a knowledge system which can (and should) be deployed across one or more servers. At its core, the Lotus Discovery Server integrates code from Domino, DB2, InXight, KeyView and Sametime. It collects, analyzes and identifies relationships between documents, people, and topics across your organization. It stores this information in a DB2 data store and presents it for browse/query through a Web interface which is referred to as the K-map.

The Discovery Server regularly updates the K-map by tracking data content, user expertise, and user activity. It gather information from Notes databases, Web sites, and file systems using spiders.

The K-map Editor interface provides access to the K-map. Internally, the Lotus Discovery Server maintains and updates the K-map through components called Discovery Services. The K-map Editor is used as an administration tool by a small set of designated editors.
Knowledge map (also referred to as K-map)
The K-map is the backbone of the Discovery Server. Physically it exists as a DB2 data store that contains content from many disparate data sources, and value-added data derived from that content. The K-map content hierarchy is often referred to as a taxonomy.

Taxonomy
A taxonomy is a generic term used to describe a classification scheme, or a way to organize and present information. The K-map is a taxonomy. It is a hierarchical representation of content organized by the K-map Builder process. An example of a taxonomy can be found at most major Internet search sites where information has been categorized into a few high-level categories.

Each top-level category contains other subcategories, and each subcategory can be further divided into other subcategories, and so on. For example, if a user selected Finance as a top-level category, he might be presented with a list of subcategories related to finance from which to further select, or refine, his search.

Knowledge map Editor (K-map Editor)
The K-map Editor is defined here as the software interface that is provided as part of the Lotus Discovery Server. K-map editors (i.e., people) use this interface to access and modify the organization of K-map content and category fields to meet the needs of the organization.

Discovery Server Control Center
The Discovery Server Control Center is the Web browser-based user interface which administrators use to configure and maintain the Lotus Discovery Server. Administrators use the Startup view to get started during initial implementation, and the Maintenance view to perform ongoing tasks, such as reallocating services and spidering new data repositories.

Discovery Services
Discovery Services are Lotus Discovery Server components that capture, analyze, process, calculate, build, and maintain information in the K-map. Following is a list of the Discovery Services:

- Spiders
- Profile Source
- Profile Synchronizer
- Metrics Collector
- Metrics Processor
- K-map Builder
- K-map Indexer
A detailed description of each of these services is provided in Chapter 3, “Components and architecture” on page 37.

**Spider**

A spider is a process used by LDS to extract information from corporate data repositories. Three types of spiders exist to support the following data types:

- Notes (includes QuickPlace, Domino.Doc)
- File system (Windows or Windows-compatible e.g. NTFS, FAT, FAT32, and also any other file system that Windows can see, such as a mapped Novell or UNIX share)
- Web

“Spidering” is the act of extracting data from these data types and converting the output into XML format for further processing by Discovery Services.

**Data repository**

A data repository is defined as any source of information that can be spidered by the Discovery Server. The following data types are supported:

- Notes databases
- QuickPlace
- Domino.Doc
- Windows or Windows-compatible file systems (e.g. NTFS, FAT, FAT32, and also any other file system that Windows can see, such as a mapped Novell or UNIX share)
- Web (HTTP)

**Training set**

A training set is a special subset of data repositories which are used to build the first draft of the K-map. Typically, these data repositories contain information which is inherently representative of the type of data the organization would like to see exposed in the taxonomy.

**Affinities**

Affinities are calculated by the Discovery Server metrics processes, which collect and calculate data about user interactions with all documents clustered within a K-map category. If the user has authored, read, edited, responded to, or created links to a number of documents in a category, then the metrics component calculates the strength of those interactions relative to others in the organization.
and proposes an affinity to the user. Approved or published affinities are stored in the user’s profile. Therefore, an affinity is really a relationship between a person and a category that already exists within the K-map. They do not show expertise, but simply an interest in, or relationship to, the topic covered in that category.

**User profile**
User profiles are initially extracted from an authoritative people source (e.g. Domino Directory from a Domino server, or an LDAP server such as IBM Secureway Directory, iPlanet or Active Directory), and then loaded into a separate Lotus Notes database (people.nsf) on the Discovery Server. Each person that the Discovery server knows about has a separate profile that is created and automatically maintained. Users can query this database directly to locate experts by skill, experience, project, education, and job type.

**Document valuation**
Document valuation is basically tracking the value of content within an organization. For instance, has the content been edited or responded to? Have doclinks been created by users to particular documents (which would indicate a higher value of that content than other documents)? Do users alter their documents? Do they create category links to documents? Do they respond to documents in a discussion database? Do they delete documents?

All these actions add up to defining which documents the people within an organization value the most. The Discovery Server metrics service does the accounting and the filing for you.

### 1.2  A first look at the Lotus Discovery Server

This section is targeted to first-time users of the Lotus Discovery Server who need to get up to speed quickly on the product. Readers who are already familiar with the product may wish to skip this overview.

#### 1.2.1 System overview diagram

Figure 1-1 on page 6 shows a typical Lotus Discovery Server system. At the top right is the Discovery Server with its K-map, people, and index databases. This system can create or add to these databases by searching though the data source examples shown on the left side of the figure. As data is added to the system, it can be managed by the K-map editors or searched by end users.
1.2.2 A guided question and answer tour

Now that you are familiar with the “essential” terminology used in describing the Lotus Knowledge Discovery System, the following question and answer tour of the Lotus Discovery Server is presented to give you a basic introduction into how the product works from a technical perspective. The objective is to put the inner workings of the Lotus Discovery Server into context, and to provide you with a foundation for the following chapters, which describe the Lotus Discovery Server components in more detail.
The Discovery Server is the back-end of the Knowledge Discovery System. Its services (the spiders, K-map building service, K-map Indexing service, and Metrics services) access, manage, and analyze the information from a variety of corporate and external sources. The Discovery Server's user interface provides search and browse access to information from these sources.

What is the Lotus Discovery Server and how does it work

The Discovery Server is a back-end server that spiders documents and your organization's directory to create a K-map (or taxonomy) of documents, and identify expertise areas of profiled users and places that the end user can browse and search.

The Discovery Server has automated tools to help with the creation of the K-map. Initially, a first draft of the K-map is created by spidering a representative selection of documents or database; this is often referred to as “selecting the initial training set”. Taxonomists (or K-map editors) then edit the initial K-map using a tool called the K-map Editor to edit the machine-generated titles and document structure (clusters) to build a working taxonomy.

Another component of the Discovery Server is metrics. This is a set of computational tasks that collect usage information and calculate document and affinity to content (expertise) ratings. First, it calculates a value for the document based on the words contained in it, and combines that with the value of document to others as tracked through metrics. Metrics then calculates an affinity between a person and documents by monitoring user interactions to content.

When that activity calculates a strong affiliation with a K-map category area, the Discovery Server generates an e-mail to the profiled individual, suggesting this affinity to K-map content. Users get the opportunity to approve or reject proposed affinities before they are published in the K-map. The published affinities help produce the expertise affinities that are published in a user's profile.

What is a user profile and how are affinities discovered

When you first set up a Discovery Server, data for user profiles is initially extracted from an authoritative people source (e.g. Domino Directory from a Domino server or an LDAP server), and then loaded into a separate profile database (people.nsf) on the Discovery Server. Users can query this database directly to locate experts by skill, experience, project, education, and job type.

As documents are processed by the Discovery Server, the metrics system collects usage information and calculates affinities between people and the documents they used and authored. The metrics system continues to automatically monitor these interactions and maintain affinities to category areas as new documents are processed and affinities are approved for publishing in a
user's profile. This has the benefit of automatically tracking areas of expertise as they change—perhaps declining in one area and strengthening in others. The metrics service also keeps track of what users do (read, doclink, forward, etc.) with the documents.

**Note:** Affinities are proposed based on relationships between people and categories in your K-map/taxonomy, and not just on a set of keywords.

### How are metrics collected from user activity

Initially, the Discovery Server extracts the raw usage data, or metrics, from spidered data sources to create document values and to create affinities. The Discovery Server continues to track user activity on documents categorized in the K-map. Document values represent the calculated sum of all user activity on the document (such as citations, forwarding, response documents, reading, etc.) and indicate a document's general value to the organization. In the K-map interface, documents are listed by this value metric.

Affinities are calculated by the Discovery Server metrics processes, which collect and calculate data about user interactions to all documents clustered within a K-map category. If the user has authored, read, edited, responded to, or created links to a number of documents in a category, then the metrics component calculates and proposes an affinity to the user.

Administrators can set a system-wide threshold (for example, calculate a 60% strength of affinity to a K-map category before proposing it to the profiled user) which controls the strength of the affinity based on a user's level of interaction in a category relative to all other users being tracked. A unique feature of the Discovery Server is that affinities are dynamic; they decay with time and inactivity.

### Do users have control over publishing of affinities

The answer is yes. The design of the product emphasizes the control of the end user in making any information available to others in their profile. For example, when the individual's interaction with documents in K-map categories reaches a designated threshold, the Discovery Server sends an e-mail notification to the end user with that proposed affinity information. The e-mail notifies the end user of the affinity determined (proposed) by the Discovery Server and requests confirmation and approval or disapproval to publish the identified affinity.

If approved, the affinity appears in the user's profile document, accessible to others through K-map search. In that way, the end user has complete control over what information indicating their expertise is published to their profile and made available to others searching for subject expertise.
Once an affinity between a person and a category has been accepted, that person is then added into the appropriate category in the K-map. People with a category affinity are ranked by the strength, and this can be easily viewed via K-map interfaces like K-station and the K-map editor.

**Note:** While the default configuration of LDS is to ask users to approve affinities, it can be configured to auto-publish without asking them.

**Where does full-text search fit in (or can I search the K-map)**

The Discovery Server includes a K-map Indexing (sometimes referred to as a full-text search) module which automatically indexes spidered repositories and produces a K-map Index. The K-map Index uses many of the features found in another Lotus technology called Domain Search, and is useful for users who know their subject very well and can enter precise search requests to retrieve it. A user can search for people who know about a particular topic, documents authored by a specific person, categories about a given topic, and so on, in addition to finding documents that contain the query term.

**How does the Discovery Server provide access to the K-map**

User access to the K-map information is authenticated using HTTP authentication. Furthermore, ACL information is retained, ensuring that K-map users only have access to documents they are authorized to review from their source location.

There are two methods of accessing the K-map: either as an editor, or as a user. For K-map editors using the K-map Builder and the K-map Editor, taxonomists (or human indexers) can categorize large numbers of documents relatively quickly to produce, re-structure, and maintain the K-map. One of the most innovative features of the Discovery Server is the ability of the K-map Builder service to “learn” from what humans do; the next time the K-map Builder Service processes documents, categorization is performed using the new rules of the K-map.

For a user, the K-map is accessed via Internet Explorer 5.01 or higher Web browser.

**Does the Discovery Server use any Domino capabilities**

The answer is yes, Domino provides rich capabilities that are leveraged in the Discovery Server. Domino environments typically have the richest content and the most consistent metadata, like titles, author attribution, etc., which makes the job of the Discovery Server much easier.
Security - what steps have you taken to ensure users' privacy

The general principle applied by the Discovery Server is that the user is always in control of the representation of himself and his information. Organizations implement Discovery Server affinity notification policies in accordance with their data privacy policies. The Discovery Server does not publish affinities or make restricted information known to the world, unless the user gives explicit permission to do that.

Specifically, end users can edit their profiles, approve or deny affinity terms for publication, and control whether their e-mail is evaluated by the Discovery Server to assess the relationship of e-mail content for affinities to the organizations.

Concerns with how the Discovery Server handles security is covered in detail in Chapter 5, “Security” on page 149.

1.3 System requirements for Lotus Discovery Server

A detailed listing of minimum hardware and software requirements can be found in the online documentation that is installed with the Lotus Discovery Server. However, as the documentation says, these are only the minimum requirements for setting up a single Lotus Discovery Server and as such, only test deployments should use these recommendations.

**Important:** Discovery Server is a vastly complex product; as such, it requires a large amount of processing power. For an effective implementation, use Table 1-1 on page 11 as an absolute minimum specification for hardware.

Estimating system requirements for the Lotus Discovery Server requires careful consideration of how you intend to deploy your Knowledge Discovery System infrastructure, and on what scale. Decisions on how much computing power is required depend on a number of factors, many of which are site-specific. Chapter 2, “Planning a Lotus Discovery Server deployment” on page 13, goes into more detail on the issues that need to be considered in deploying the Lotus Discovery Server.

1.3.1 Suggested hardware requirements. Use with caution!

Table 1-1 attempts to quantify server hardware requirements for three different types of deployment of the Lotus Discovery Server.
**Note:** System administrators are urged to use the following recommendations with caution as there is no guarantee the requirements will be suitable for your site or scale of deployment.

<table>
<thead>
<tr>
<th>Deployment option</th>
<th>No. of servers</th>
<th>Suggested hardware (minimum)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Demonstration</td>
<td>1</td>
<td>Single processor 500Mhz PC</td>
</tr>
<tr>
<td></td>
<td></td>
<td>20 gigabytes disk capacity</td>
</tr>
<tr>
<td></td>
<td></td>
<td>512 Mb physical RAM</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1024 Mb virtual RAM</td>
</tr>
<tr>
<td>Pilot</td>
<td>2</td>
<td>2 x Dual Processor 800MHz Server</td>
</tr>
<tr>
<td></td>
<td></td>
<td>50 gigabytes disk capacity</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1024 Mb RAM</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2048 Mb virtual RAM</td>
</tr>
<tr>
<td>Production</td>
<td>4</td>
<td>2 x Dual Processor 800MHz Server</td>
</tr>
<tr>
<td></td>
<td></td>
<td>50 gigabytes disk capacity</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1024 Mb RAM</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2048 Mb virtual RAM</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2 x Quad Processor 800MHz Server</td>
</tr>
<tr>
<td></td>
<td></td>
<td>100 gigabytes disk capacity</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2048 Mb RAM</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4096 Mb virtual RAM</td>
</tr>
</tbody>
</table>

**1.3.2 Startup versus ongoing requirements**

Keep in mind the different system requirements at start-up and on an ongoing basis—it is easy to forget that some of the performance issues at start-up do not apply after the system is up and running. For example, spidering is a network-intensive operation the first time a data repository is spidered. Where possible, server startups should be scheduled out of prime hours to minimize the impact on users. The same scheduling requirements could also apply to spidering of large new data sources, or re-spidering of existing data sources.

**Tip:** Don’t be surprised by the amount of processing needed when initially setting up a Discovery Server; the first few days can completely overload a system as it works to build its internal relationships. However, this will reduce over time.

Once a data repository has been spidered completely, further spidering should settle down to steady state where data extraction is performed on an incremental basis. Stress on the Lotus Discovery Server should also be correspondingly low.
Planning a Lotus Discovery Server deployment

The deployment of any product is made easier through careful consideration of relevant issues and thorough planning before rollout. By taking time to consider how the product is going to be used, analyzing where you currently stand, and planning how you are going to get where you want to be, you can make a major contribution towards a successful rollout of the Lotus Discovery Server within your organization. Administrators will find building and maintaining a Lotus Discovery Server environment involves a great deal more planning than most other products because it spans technical, organizational, and cultural issues that must be taken into account as part of the planning process.

This chapter is intended for anyone who has the task of deploying a Lotus Knowledge Discovery System (KDS) in their organization. You are encouraged to look at your deployment analytically, and there is also advice on how to build an effective deployment team, the importance of managing your data, and a discussion of strategies which can help you achieve a successful deployment of your Lotus Knowledge Discovery System.
2.1 Getting started

The commissioning of a knowledge management initiative is a serious undertaking that requires careful planning and allocation of resources to ensure a successful implementation is the end result. Project managers will find they have to deal with a diverse, and at times complex range of issues which span organizational boundaries and involve people from various parts of the company.

A key strategy in managing a major project like deploying the Lotus Discovery Server is to break up the project into manageable stages and sub-projects. A good starting point is to first define the problem you are trying to solve, then look at where you are currently, and then decide how you will get there. Table 2-1 presents a top-level breakdown of how you might initially approach the problem:

<table>
<thead>
<tr>
<th>Table 2-1  Three stages of awareness</th>
</tr>
</thead>
<tbody>
<tr>
<td>Articulate your knowledge management strategy</td>
</tr>
<tr>
<td>Conduct a Knowledge Audit</td>
</tr>
<tr>
<td>Prepare a project plan</td>
</tr>
</tbody>
</table>

2.1.1 Articulate your knowledge management strategy

A large number of knowledge management initiatives fail to deliver expected benefits because the organization has failed to consider the complex interplay of technological, organizational and cultural issues involved. Ideally, the basis for adopting a knowledge management solution like the Lotus Discovery Server should be incorporated into the organization's strategic IT planning policy. If you are a member of a rollout team that has been commissioned to deploy the Lotus Discovery Server, try answering the following questions:

- Is the organization's knowledge management strategy clearly articulated as a corporate document for anyone to read?
- Can Lotus Discovery Server team members understand and relate to the organization's knowledge management strategy, and feel like they can take ownership of the corporate initiatives coming out of it?
- Does the Lotus Discovery Server deployment project have the support of senior management?

If the answer to any of these questions is negative, your chances of successfully rolling out the Lotus Discovery Server are significantly reduced. Make sure your organization is not simply jumping on the knowledge management bandwagon, without stating why it is doing so.
2.1.2 Conduct a knowledge audit

Conducting a knowledge audit is a major step towards understanding what issues you need to confront as part of your project plan. By now, you should be aware of the basic hardware and software requirements of the Lotus Discovery Server (refer to 6.3, “Server hardware considerations” on page 187 for more information on this). However, you will also need to be aware of cultural and organizational issues that come into play as part of the Lotus Discovery Server deployment. To assist you in creating your deployment plan, the following list of questions can be used to conduct a knowledge audit of your organization.

What problem is your organization trying to solve

Are you interested in: facilitating the identification and use of corporate expertise (expertise location); leveraging knowledge from disparate sources of data (knowledge discovery); making more efficient of existing knowledge (more recycling of knowledge assets, with less “reinvention of the wheel”)?

To some extent, the answer can be drawn from your organization’s knowledge management strategy (assuming you have one). However, the question should still be included in your audit as useful information can be collected on how well end-user responses align with corporate expectations.

What information assets do you currently have

Identify which information repositories contain data that has value to the organization. The Lotus Discovery Server can spider data stored in:

- Lotus Notes databases (including Domino.Doc, QuickPlaces, K-station places)
- File systems
- Web sites
- People Directories
- ERP or RDMS content via DECS

Exclude information stored as image files and proprietary formats which cannot be translated by the Lotus Discovery Server. For a full listing of file formats accepted by the Lotus Discovery Server, refer to Appendix A, “File types you can spider” on page 249.

This information will determine what number and what kind of spiders you can run.
What state is your current data in
Do your data repositories contain useful meta data and rich content that can be extracted by the Lotus Discovery Server to build a knowledge map/taxonomy? Lotus Notes databases offer the richest source of information in terms of meta data stored with the rich text content. Rich data sources facilitate categorization of information into clustered topics, generation of user expertise, and calculation of affinities.

When examining your data sources, check for consistency of content, particularly where data has been authored/produced using different generations/releases of software.

Note: Documents must contain a minimum of 30 unique tokens (a token is a word) to qualify for inclusion in the K-map.

Who is the intended audience
It is important to know who will be using the data and for what purpose. For example, are you providing a solution for an area involved in Customer Relationship Management, or will you take a broader, corporate-wide initiative? This information will influence the selection of data sources you spider to build and populate your K-map.

Where is your information stored
Typically, the Lotus Discovery Server is located on a high-speed, centralized part of the network. Data stored on slow network segments, restricted/secured networks and on the other side, WAN links, may not be accessible.

Note: Even if access across the WAN is possible, spidering of data repositories may have serious performance impacts on a slow network.

How many gigabytes of information do you expect to spider
This is likely to influence how you deploy your Lotus Discovery Server infrastructure. Keep in mind the difference between initial startup and ongoing demands of the Lotus Discovery Services. For example, first-time spidering a data repository with 130,000 documents can take a while because all documents are considered new. Subsequent processing of the same data repository will be much faster if only a handful of existing documents are modified and/or created each day.
When conducting your audit, useful attributes to collect per data repository are:

- Number of documents
- Total size of repository
- Number of attachments by size and type

The information you collect will provide input into the following:

- How much disk storage is required
- Capacity of existing network to handle spidering of data repositories
- When to schedule spidering of repositories
- Estimates on how long it takes to spider data repositories

**Who owns the information and is it accessible**

Consult with data custodians to make sure their data repositories can be spidered by the Lotus Discovery Server. Where access to information is restricted, information should be compiled on:

- Repository name, type and location
- Who owns it
- Why access is restricted
- If partial access is allowed, identify what parts
- How long access is restricted

It is important to maintain the accuracy of this information, as you don’t want to accidently expose information that has not been correctly secured in its native environment (e.g. incorrect ACL settings on a Lotus Notes database).

**How much information is stored in user mail files**

Some organizations may have large amounts of corporate knowledge stored in the mail files of users. Investigate how much and what type of information is inaccessible to organization.

**Tip:** Add up your total disk storage in megabytes and work out what percentage of this is stored in user mail files. This figure may surprise you!
Who will edit and maintain the K-map
Initially a member of the Lotus Discovery Server deployment team who has expertise in classification/hierarchy/taxonomy concepts may build the K-map. Identification of users who have the necessary expertise in managing the K-map is an important task because these people will be responsible for maintaining the backbone of the Lotus Discovery Server - the K-map.

What is your organization’s policy on privacy
If you intend to spider mail files, you will need to develop a specific policy and announce it to users. Assess user reaction to this idea as part of your audit.

Will you publish user affinities with or without consent
It's also important to determine how and if you will publish personal affinities. Assess user reaction to this idea as part of your audit.

2.1.3 Create a project plan
After conducting a knowledge audit and making sure your organization has a clearly articulated knowledge management strategy, you are now ready to create a project plan that defines how the project team will deploy the Lotus Discovery Server. An important piece of advice at this stage is “be realistic in deciding what you can achieve with the resources and time you have available”.

The following checklist highlights some of the issues to consider when creating up your project plan. Remember to include site-specific issues (for example, your company's IT security policy).

▸ Determine the scope of the project.
▸ Break up (or downsize) the project into manageable segments.
▸ Match the resources available to the size and complexity of the project.
▸ Target functional areas that have a clear, focused business objective with well-defined benefits.
▸ Avoid multi-functional areas that have multiple objectives and offer intangible benefits; such areas can easily scuttle your Lotus Discovery Server rollout through lack of progress and unproductive effort.
▸ Determine what size pilot you are capable of supporting with the resources available.
▸ Include education and promotion as part of your project plan.
▸ Allow for production of status/management reports.
▸ Identify critical stages and high-risk factors in the project and propose how they will be managed.
Create timetables that can be used to monitor project status.
Identify key players and define communication channels between them.

Following is a proposed framework for creating a Lotus Discovery Server deployment plan. This can be used as a starting point for drawing up your own project plan.

### 1. Prepare a business case

A business case may already be written for you, or exist as part of your organization's strategic forward work plan. In any case, this document seeks approval for and commitment from management to support your proposal to roll out the Lotus Discovery Server to the organization. This is usually where estimates for resources (staff, hardware, training, promotion, etc.) are submitted, and where approval for funding is sought from management.

### 2. Assemble a multi-disciplinary team

Knowledge management initiatives usually involve many areas of your organization. Implementation of the Lotus Discovery Server requires more of a commitment than rolling out a simple desktop application. Planning a deployment brings into play a range of organizational and technical issues which involve training, promotion, system administration and design, research, and so on. In some organizations, the task of establishing a team with the necessary skills may be performed by a Chief Knowledge Officer (CKO). Others will need management support to form a new team from existing staff resources.

### 3. Build a test/demonstration environment

This stage achieves the following objectives:

- Enables team members to become familiar with the product
- Allows proof of concept tests to be conducted
- Allows the product to be demonstrated to key players such as line of business managers
- Helps identify technical, cultural and organizational issues not previously anticipated
4. **Prepare a project plan for a pilot study**

This stage defines the parameters for conducting a pilot study in your organization. The scope, people, data, project objectives, anticipated benefits, dates, support arrangements, training requirements and promotion material are documented in a detailed project plan. Critical success factors for a pilot are support from management, clearly defined objectives, careful allocation and management of resources, selection of a representative user base, and capture of meaningful information for analysis.

5. **Conducting the pilot**

Make sure you have appropriate recording mechanisms in place for measuring, monitoring and managing the pilot. Be prepared to modify or suspend the pilot if the project outcomes are not being realized or significant problems arise. Remember: the pilot is (normally) meant to be a trial run for a full-scale deployment.

6. **Conduct a post mortem on the pilot**

Consider the following questions:

- Was the pilot a success?
- Did it provide sufficient information to confirm you are ready for a full scale deployment?
- What did you learn?
- Are your resources estimates (people, hardware, training material) for a full deployment correct?

7. **Prepare for a production release**

Integrate lessons learned from the pilot into the final rollout plan. Acquire and put in place infrastructure to support a full-scale rollout of the Lotus Discovery Server. Adjust server sizing estimates to meet projected load. Commit timetables for production release and advertise the implementation schedule. Conduct training and corporate awareness seminars to create the desired mindset and promote shifts in organizational and cultural behavior. Update the project plan for a full-scale deployment.

8. **Production rollout**

Make sure the support infrastructure is ready to accept the challenge. Post regular updates on the progress and uptake of the product. Manage problems and respond to feedback on implementation and usage of the product.
2.2 Selecting the right people for the deployment team

Planning and rolling out a knowledge management system solution using the Lotus Discovery Server requires considerable consultation and collaborative effort between technicians, line-of-business managers, end users, data custodians, librarians/taxonomists, and the executive. Lotus recommends establishing a multi-disciplinary team which includes representatives from various areas and levels of responsibility to guide a successful deployment of the Lotus Discovery Server. A team built on such a foundation will be better equipped to deal with the technological, organizational and cultural issues that come into play when designing a knowledge management solution for the enterprise.

Some organizations already have a Chief Knowledge Officer (CKO) in place to direct and coordinate resources for knowledge management initiatives. In this case, the CKO can help establish a team with the right personnel to run the knowledge management project. Other organizations will need to put together a team which has support and commitment from senior management or the Chief Information Officer. In any case, the team must have executive support to accomplish implementation of a major knowledge management initiative like the Lotus Discovery Server.

2.2.1 Composition of a Lotus Discovery Server deployment team

Lotus has identified a number of roles within a typical Discovery Server implementation that describe the type of work people in the organization will be performing. In looking at these roles, you can identify attributes and skills that would be useful for team members to possess. Keep in mind the size of your team is likely to grow as you progress through various stages of

9. Post implementation report

The final stage involves conducting an “honest” assessment on the success of the rollout. This may involve running a survey to collect end-user feedback and interviewing line-of-business managers to ascertain whether anticipated benefits have been realized. Bring the rollout team together for a post-rollout analysis and assessment of the following:

- What benefits has the organization realized following the rollout of the Lotus Discovery Server?
- What remains to be done?
- Is the organization’s knowledge management strategy still correct or on track?
- Have desirable shifts cultural and organization behavior been achieved?
implementation—from initial proof of concept and demonstration trials—through to conducting a pilot and eventually, final deployment. Team members may be able to cover multiple roles depending on the skills they possess and the number of people available.

**Tip:** Sharing roles between team members encourages sharing of information and reduces the impact of staff turnover on the remaining team members.

**Sponsor**

This person could well be the Chief Knowledge Officer or at least someone who is involved at a management level. The person is likely to be someone who has been with the company for a number of years, knows the organization well, and can act as a “visionary” who is able to articulate the organization’s knowledge management strategy. In brief, the sponsor oversees the entire Lotus Discovery Server project and is probably not directly attached to the team. It is very unlikely that this person will be directly involved with setting up and configuring the Lotus Discovery Server.

As part of this role, the sponsor is usually responsible for:

- Establishing the scope of the deployment.
- Managing the deployment and administration of team effort.
- Keeping the executive informed on the status of knowledge management initiatives.
- Providing strategic advice
- Giving input to funding and allocating team resources
- Selecting team members

**General skills which are assumed or required:**

- Understands the value of information to the business the company is in
- Can influence senior management and has respect within the organization
- Understands the company culture
- Possesses interdisciplinary knowledge (knows who does what within the company)
- Possesses overall knowledge of the type of data/data sources available
- Possesses overall knowledge of data security policy
- Knows who to talk to in order to get that information
**Metrics evaluator**

This person uses metrics to assess what (and how) information flows into, out of, and within the corporation. Within the Lotus Discovery Server deployment team, the role of this person will change depending on what stage the deployment is at. Initially, this person will be concerned with validation and analysis of metrics data to ensure usable information is being generated.

Later, as the metrics evaluator becomes familiar with and gains confidence in compiling information and producing reports using the metrics data, the role will focus more on how to enable the organization to make best use of metrics data. In the final stages of deployment, this person is likely to be working closely with a wide spectrum of managers to demonstrate how to make effective use of metrics data.

Table 2-2 profiles the types of people the metrics evaluator may want to contact.

<table>
<thead>
<tr>
<th>Job responsibility/title</th>
<th>Information these people are interested in</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOB managers, Department/Division/Group managers</td>
<td>▶ What do the people that report to me need?</td>
</tr>
<tr>
<td></td>
<td>▶ Who in my group is contributing/authoring?</td>
</tr>
<tr>
<td></td>
<td>▶ What knowledge is relevant to my group/department?</td>
</tr>
<tr>
<td></td>
<td>▶ What are “thought leaders” in my group reading, so I can have the other people in my group read those things?</td>
</tr>
<tr>
<td>Systems architects, systems managers, application managers</td>
<td>▶ Who is using which systems/repositories (i.e. Exchange publish folders, Notes DBs, etc.)?</td>
</tr>
<tr>
<td></td>
<td>▶ Who is contributing?</td>
</tr>
<tr>
<td></td>
<td>▶ Are there problems with any of the systems?</td>
</tr>
<tr>
<td></td>
<td>▶ Do the right people have access to the data they need?</td>
</tr>
<tr>
<td></td>
<td>▶ Which databases are active vs inactive?</td>
</tr>
</tbody>
</table>
General skills which are assumed or required

- Can analyze data to reveal trends, relationships, and patterns in that information.
- Possesses business process knowledge (knows how data is used to manage the business)
- Possesses interdisciplinary knowledge (knows who does what in the company)
- Possesses overall knowledge of the type of data/data sources available
- Understands the value of information to the business their company is in
- Understands the company culture
- Can articulate (through information sharing forums) how metrics can be used
- Possesses an affinity for wanting to help users do their job better
- Possesses competent (or advanced) computer skills

Taxonomy editor

This person could be a librarian or classification expert who is seconded to the team on a full-time or part-time basis. The taxonomy editor is responsible for overseeing the classification of information in the organization's K-map. This person is expected to provide specialist advice into the initial building and tuning of the K-map, and a reasonable level of technical competence is desirable. Other duties for this role are:

- Selecting suitable data sources for building the organization's initial K-map
- Labelling/relabelling of categories
- Re-training the K-map
- Classifying documents
- Identifying relevant data sources for populating the K-map
One of the main challenges of this role is making sure the resultant K-map meets the business needs of the organization and is consistent with the scope of the project.

**General skills which are assumed or required:**

- Possesses a solid understanding of classification/hierarchy/taxonomy concepts
- Possesses business process knowledge (knows how users interact with data and how it is used to manage the business)
- Possesses interdisciplinary knowledge (knows who does what in the company)
- Possesses a sound knowledge of the type of data/data sources available
- Works comfortably with graphical user interfaces like Windows Explorer or equivalent, and understands conventions like drag and drop, cut and paste, renaming, and so on
- Possesses knowledge retrieval skills (competent in constructing advanced search/find queries)
- Possesses sufficient technical competency to understand how the Lotus Discovery Server works (including concepts like affinities, clustering, and document scoring)

**Server administrator**

This person is responsible for setting up and managing the infrastructure required for deploying the Lotus Discovery Server. Duties include installing and configuring the Lotus Discovery Server software, enabling the connections to the requested data sources, tuning and system maintenance, monitoring system performance, and providing technical advice. Effective execution of this role is crucial to the success of the Lotus Discovery Server deployment.

This person is likely to be someone who is an experienced Domino Administrator, who knows the IT infrastructure well, and has a strong background in managing complex “startup” projects. One of the main challenges of this role is dealing with the interplay of technical, organizational, and cultural issues that arise in a Lotus Discovery Server implementation.

**General skills which are assumed or required:**

- Possesses advanced knowledge of Domino administration
- Possesses advanced knowledge of Windows NT/2000 administration
- Possesses sound knowledge of the organization’s IT infrastructure
→ Is familiar with major ERP and CRM systems/applications used in the organization
→ Possesses knowledge of IT security policy
→ Possesses knowledge of the organization’s e-mail and document management policy
→ Possesses knowledge of the infrastructure used for collaboration in the organization (Lotus Notes, MS Xchange, Intranet)
→ Possesses knowledge of DB2 (or other RDBMS) administration function; is (or has been) a database administrator (DBA)
→ Possesses an understanding of taxonomy/classification concepts
→ Possesses a sound knowledge of the type of data/data sources available
→ Understands the value of information to the business their company is in

**End user**

Representation of end-user interests is usually performed through regular consultation with users, and through seeking feedback from users across the organization. In some cases, recruitment of users with specialist knowledge or skills may add a valuable dimension to the deployment team which could not otherwise be covered by existing team members. For example, conducting promotional activities and education programs may well be best serviced by users who have proven presentation and training skills.

**Tip:** A short-term assignment of selected users into the deployment team can provide valuable business perspective and context to team members.

**General skills which are assumed or required:**

At all stages of the Lotus Discovery Server deployment, end users should be encouraged to do the following:
→ Provide feedback on usability and performance
→ Report problems on access, crashes, hangs, errors, etc.
→ State expectations
→ Actively participant in pilots, trials, etc.
→ Participate in, and contribute to, corporate programs and initiatives
→ Participate in establishing rollout schedules
→ Attend education programs and training sessions
2.3 The importance of data management

The information stored in your organization's data repositories is the primary foundation of your knowledge management assets. The following sections deal with data management issues that most organizations will be familiar with and manage, with varying degrees of success. In working through these issues, it will become clear that a key indicator of an organization's preparedness for implementing a knowledge management solution is the existence of a clearly articulated document management policy—and the extent to which the policy has been put into practice.

2.3.1 What state is your data in

A major result of a knowledge audit is the identification of your organization's information assets. But how much of this information is stored in a format that can be effectively used? It's a good idea to check and score your data repositories according to the amount of meta data elements and rich text content stored in the documents/files. For data repositories which attract a low score (i.e. little or no useful meta data or poor content), consider the following options:

- Running a program to attach meta data elements
- Reformatting the data into a usable state
- Modifying systems which create the data, to ensure the meta data is being created and updated correctly
- Retraining authors to add meta data elements value to their work
- Changing corporate culture/attitudes to promote authorship and capture of better quality data

Tip: By ranking and scoring the quality of your data repositories, you can identify weaknesses in the capture of information. This effort also helps the Lotus Discovery Server administrator determine which data repositories effectively contribute to the production of a good K-map.

Note: The deployment team should ensure that mechanisms are put in place to facilitate user consultation and feedback.
2.3.2 Dealing with legacy data

The decision to process legacy data (for example, several years of Customer Relationship Management data entered via a legacy data entry system/application) will largely depend on the quality of the information captured and what format the data is stored in. Large volumes of information are likely to be involved. If possible, processing this type of data should be done on an “informed” basis. Consider the following issues:

- Is the data content relevant?
- Will large volumes of essentially out-of-date data skew the content of the K-map in a way that obscures useful information?
- Can the information be preprocessed in a test environment to verify its value?
- Can the system handle the amount of information available?
- Does the data add value to your knowledge management solution?

2.3.3 Start managing your data today

An analysis of your knowledge audit should reveal what you are doing well and not so well. For areas which have been identified as problematic, the following questions may help you identify the source of the problem:

- How effective has your organization been in implementing practices which are stated in your data management policy?
- Is your data management policy deficient/out-of-date/non-existent?
- What obstacles are getting in the way? For example:
  - Is corporate culture the problem (e.g. lack of information sharing)?
  - Is your software capable of producing what you want?
- Do users have:
  - The necessary skills and tools available?
  - Access to non-standard software?
- Are IT security policies too loose or too restrictive?

Managing your data really starts with having an up-to-date, relevant data management policy in place. Organizations who have invested effort in getting this right are likely to be well positioned for implementing a Knowledge Discovery System solution.
2.3.4 Implement standards

An important aspect of document management is implementation of document standards. In its most visible form, this applies to creation of information at the time of authorship. In a tightly controlled environment, authors may be asked to supply mandatory meta data before their work can be saved in a managed data repository.

This is good for subsequent data reuse, but is invariably unpopular with end users if the overheads in saving their work are high. Where possible, user software should remove such burdens from the user to encourage appropriate capture of meta data which meets required standards. End users can be remarkably adept at finding ways around cumbersome user interfaces, which works against your efforts to enforce standards for capturing information.

The answer is to strike an acceptable balance between authorship obligations and system requirements to meet standards. More sophisticated front-end software can reduce this overhead through automatic capture of previously declared user and system attributes (e.g. settings for document expiry date, security, filing, authorized readers/editors, and so on).

Another side of implementing standards is enforcing consistency across information sources. The ability to draw disparate sources of information together can be quickly compromised if your data is stored in a variety of formats which cannot be easily correlated.

2.3.5 Electronic cleanups

Accumulation of large amounts of information in electronic data repositories is pretty much a fact of life for most organizations. Technology allows end users to rapidly collect and distribute information, and disk storage costs per megabyte seem to reduce at a rate which allows an organization to accommodate the growth at about the same net cost.

In many cases, organizations simply adopt a policy of purchasing more disk to meet demand. Many users simply hoard information on the basis of “I might need it” or “I don't have time to clean it up”. The problem is that there are a number of hidden costs in adopting such an approach which can seriously degrade the performance of your Lotus Discovery Server. The hidden costs are:

- Processing time required to extract information from data repositories
- Productivity lost by users trying to locate the right information at the right time
- Performance impact when trying to access and maintain the system
The “up-stream effect” where higher speed networks, larger computers, more storage and larger capacity backup solutions are required to host an ever-growing pool of information.

Factors working against implementing a cleanup policy are:

- A non-existent (or out-of-date) data management policy, or non-compliance with a data management policy.
- Failure to attach suitable attributes (meta data) to the data at the time of authorship.
- The cost per megabyte of disk storage keeps getting cheaper.
- Inbound data exceeds the organization’s capacity to assimilate it.

**Beware of data loss when you clean up**

Old information doesn’t simply translate into useless information that can be thrown out purely on the basis of a cutoff date. Any piece of information can have inherent value that effectively contributes to tacit discovery of knowledge. For example, how often have you compiled a report or information paper, only to find later that someone (or several people) has previously produced similar material? Valuable material is often left unused and forgotten, leading to wasted effort and reinvention.

When it comes to cleaning up large data holdings to manage disk space, many organizations employ purging practices based on simplistic criteria like date created, last modified, or even size. Such practices can result in significant loss of valuable corporate knowledge. The problem is, some organizations simply have no choice because they have no way of determining what is valuable and what is not, without investing considerable time and effort to assess the value of the information.

**Tip:** Declare an “electronic cleanup” day where users are asked to throw out the electronic garbage (according to your guidelines). Schedule two or three times a year.

### 2.3.6 Customizing your environment - the hidden costs

Customizing your IT environment can seem like a good idea, but in the long term it can result in unexpected costs to the organization, particularly in the areas of data management and knowledge management. A good example of this is customization of the Domino Directory. Out of the box, the Domino Directory is designed to interoperate with a wide range of software including the Lotus Discovery Server. Modification to the standard design could degrade the ability of the Profile Source spider to extract user profile information.
The same applies to the following Notes templates, which have known data field definitions that are recognized by default by the Notes spider:

- Discussion (StdR50Disc)
- Teamroom (StdR50TeamRoom)
- Mail (StdR50Mail)

Creating data repositories based on standard Notes templates can also deliver the following benefits:

- Compatibility with Lotus Discovery Server and K-station, both now and in the future
- Consistency between data repositories
- Free upgrades to new features
- Lower support costs

### 2.3.7 Selecting data sources

Conducting a knowledge audit can reveal a great deal of valuable information about your knowledge assets and preparedness for implementing a knowledge management solution. Implementing an appropriate data management policy can also contribute greatly to your organization's ability to implement a knowledge management solution like the Lotus Discovery Server.

When it comes to selecting data sources for building and populating the K-map, the Lotus Discovery Server Team will be better equipped to identify what information assets can be processed to meet the objectives of your knowledge management project. Keep in mind that rich text data repositories with good meta data yield the most suitable content for the K-map. Organizations which already have a Domino solution in place are probably best positioned to deploy a Lotus Knowledge Discovery System solution.

When selecting data repositories, target good quality data repositories first. In terms of repository types, spidering Notes databases is likely to produce the best output, followed by file systems and then Web sites.
2.4 Strategies for success

Building and sustaining a knowledge management solution can challenge even the most successful organization. Knowledge management readiness requires careful consideration of issues before rollout. It is important to have the right personnel on your Lotus Discovery Server deployment team, to plan out your implementation strategy and have commitment from the organization as a whole. This section discusses strategies that can help you achieve a successful implementation of the Lotus Discovery Server.

2.4.1 Read the documentation

We cannot emphasize enough the importance of reading the online documentation that is installed with the Lotus Discovery Server software. Long after this Redbook has been published, the one source of information you can be sure is up to date is the online documentation shipped with the product. Of particular importance is the readme.chm file, which contains the latest release notes on features and issues to watch out for when installing and configuring the Lotus Discovery Server.

Table 2-3 lists the file name and title of each online manual. Unless otherwise specified, all files are installed under the Lotus\KDS\Data\Help folder.

Table 2-3 Location of online KDS documentation

<table>
<thead>
<tr>
<th>File name</th>
<th>Document title</th>
</tr>
</thead>
<tbody>
<tr>
<td>readme.chm</td>
<td>Release Notes for:</td>
</tr>
<tr>
<td></td>
<td>▶ K-station</td>
</tr>
<tr>
<td></td>
<td>▶ Lotus Discovery Server</td>
</tr>
<tr>
<td></td>
<td>▶ Lotus Discovery Server Thesaurus Tool</td>
</tr>
<tr>
<td>install.chm</td>
<td>K-station Install and Setup Help</td>
</tr>
<tr>
<td>ds_admin.chm</td>
<td>Discovery Control Center Guide</td>
</tr>
<tr>
<td>K-map Editor.chm (located in</td>
<td>K-map Editor Help</td>
</tr>
<tr>
<td>the \Lotus\KDS\folder)</td>
<td></td>
</tr>
<tr>
<td>K-station_user.chm</td>
<td>Using K-station</td>
</tr>
</tbody>
</table>
2.4.2 Obtain commitment

Experts agree that an enterprise-wide knowledge management initiative stands little chance of success unless upper management takes ownership of the project. Project leaders can initiate the effort, but commitment from upper level management to get everyone behind it is an essential ingredient. As previously mentioned, the knowledge management project should also have a “visionary”, someone with credibility in the enterprise who can articulate the vision behind the initiative and set the strategic direction.

Many knowledge management projects fail before they've even begun because the company has failed to consider the complex interplay of technological, strategic and cultural issues involved. Ideally, the basis for adopting a knowledge management solution should be incorporated into the company's strategic IT planning policy. The payoff for the enterprise that can successfully balance these concerns is transformation into a more efficient and productive corporation whose main asset is knowledge.

2.4.3 Check security

The Lotus Discovery Server needs to access key resources within your organization. Some of these—such as strategic databases—may have security controls that need to be adjusted to enable access by the server. You can be assured the Lotus Discovery Server respects your existing security controls based on the individual rights of the user—provided that you have set it up correctly in the first place. As part of your deployment plan, include an audit of data sources in your organization. For more information on security, refer to Chapter 5, “Security” on page 149.

Tip: Use the Domino Catalog to identify incorrect ACL settings for Notes databases. The Domino Administrator can quickly make up private views to expose problem areas.

2.4.4 Building your taxonomy (K-map)

The process of preparing a K-map for your organization can be a time-consuming and frustrating process for an uninitiated or underprepared project team. The length of time it takes to build an acceptable K-map for public use can range from a couple of weeks to months, depending on how much effort you have put into the following:

- Identifying the needs of your users
- Defining the problem the organization is trying to solve
- Editing and tuning the K-map
- Recruiting people who have the necessary expertise to manage the K-map
The K-map is very much the backbone of the Lotus Discovery Server. Be prepared to spend time building, revising, and if necessary rebuilding your K-map. Allow time to experiment with various data sources to learn how the system can automatically prepare an initial K-map. Make sure you audit your data sources for quality content, which promotes generation of a K-map that meets your users' needs. If you have an existing taxonomy in place in your organization, determine how users work with it and whether parts of it should be reproduced, enhanced, or discarded.

2.4.5 Establish the scope of the project

Establish the scope of the knowledge management project early and try to be realistic in what you can achieve. Is the planned knowledge management program enterprise-wide? Does it extends across multiple organizational and geographic boundaries?

Knowing the answers to these questions will provide important context to the project and enhance the likelihood of a successful implementation. Consider scaling the initial knowledge management effort to a manageable state. If necessary, do the following:

- Break up or downsize the knowledge management project into manageable segments. Match the resources available to the size and complexity of the project.
- Target functional areas that have a clear, focused business objective and well-defined benefits.
- Avoid (initially) multifunctional areas that have multiple objectives and offer intangible benefits. These areas can easily scuttle your knowledge management program through lack of progress and deliverable results.

2.4.6 Encourage the right culture

It's widely accepted that a favorable corporate culture is a prerequisite for the successful implementation of knowledge management initiatives—you have to focus not just on technology, but on culture and process as well.

Sustainable knowledge management requires the introduction of a knowledge-sharing culture organized around key skills and a set of incentives to reuse knowledge to reinforce the skills required for the effective operation of the organization. You may not agree entirely with point of view, but being aware of
what is being written about knowledge management and considering how knowledge sharing and information reuse is currently practiced in your organization may reveal the type of culture you have to work with when you draw up your knowledge management program.

A key success factor in the implementation of KDS is encouraging the right mindset in the staff that is responsible for delivering a knowledge management solution to the organization.

2.4.7 Involve, train, and inform your end users

The benefits of deploying a knowledge management solution are likely to be lost on most users in your organization because there is no immediate result you can point to. With the Lotus Discovery Server, you are rolling out new technology that requires a certain amount of education, end-user involvement in pilots, training, and promotion to ensure acceptance and effective use of the product.

In preparing your deployment plan, allocate time for the following tasks:

- Preparing users on privacy issues like spidering of mail files
- Informing users how affinities and expertise location will operate
- Updating corporate policy to reflect changes in IT practices and information access

Users will be understandably apprehensive about the generation of knowledge and information that can be associated back to individuals. It is important to allay these concerns early, in the planning stages of deployment, rather than waiting until actual deployment and risking reactive hostility towards a new product that appears to be invading their privacy.

2.4.8 Obtaining the right hardware

Enterprise commitment for knowledge management initiatives can be severely tested if inadequate hardware resources have been allocated. KDS has significant resource demands, particularly in a large distributed implementation. System administrators should resist using second-hand hardware that does not meet the minimum system requirements.

Scale hardware requirements according to the solution being implemented. A high-end 700Mhz desktop PC might be suitable for running a demonstrator or proof of concept Discovery server, but reusing the same hardware for a multiuser pilot could jeopardize the future of your knowledge management program. Refer to 6.3, “Server hardware considerations” on page 187 for more information on sizing your hardware infrastructure for your Lotus Discovery Server.
2.4.9 Competing projects

Beware of deploying a major knowledge management initiative concurrently with other major programs the organization is working on. For example, deploying KDS at the same time your organization is rolling out Windows 2000 should be avoided.

This is because IT personnel will have their attention diverted away from your project, and the configuration, performance, and possibly reliability of your organization's IT infrastructure will be in a state of change. In addition, your ability to measure and monitor the performance characteristics of your Lotus Discovery Servers will be complicated by changes in the underlying infrastructure.
Chapter 3. Components and architecture

This chapter will help you understand the architecture of the Lotus Discovery Server.

We detail how the Lotus Discovery Server works and how the components interact with each other, as well as what kinds of data flows between the components and for what purpose.
3.1 What are the tasks and what do they do

The Discovery Server System is based on an interaction between six major components. The worker tasks (called Discovery Services) work on predefined data which has earlier been delivered to a back-end database by semi-intelligent robots. These robots, called spiders, work on remote and local data repositories to mine meta data about those contents. The robots and worker tasks are controlled by a manager task, called the scheduler.

![Figure 3-1 LDS general data flow chart](image)

Figure 3-1 shows a high-level view of what happens in a Discovery Server. The scheduler starts by activating a spider which writes content to an XML queue. The workers then read data from these queues, modify it, and store it to the K-map database.
3.2 Scheduler

The Discovery Server scheduler task is started automatically on the primary Discovery Server. It controls work queues and spider schedules and manages the dynamic starting and stopping of all tasks not automatically run on every Discovery Server. (This applies only to LDS tasks, not to Domino server-specific tasks like the HTTP task or the DIIOP task.)

The corresponding Discovery Server task is called dssched.exe and must never be stopped or reloaded.

**Note:** Always use the command `tell KDS shutdown` to shut down the server; do not use the quit command familiar to Domino administrators. Shut down secondary servers first, before shutting down the primary server.

The queue is populated by the scheduler, which constructs each request from the data in the Notes database called dashboard.nsf. The scheduler collects the information from the Data/Repository (which is specific and different for each repository) and from the Data/Spider Settings section of this database (which is general and the same for all spiders of that type).

In Figure 3-2 on page 40 you can see the following processes:

1. The scheduler reads spider/worker schedules from dashboard.nsf.
2. Monitor Completion queue for spider and worker start, stop, or progress messages.
3. If scheduled to run, issue start processing request to individual spider queue/worker queue.
4. Spiders/workers issue progress messages and completed messages to the completion queue. If errors occur, issue the corresponding problem report to the queue.
5. Certain non-repository-specific messages on the completion queue are relayed by the scheduler to dashboard.nsf.
6. If a spider needs to yield processing (e.g. scheduled time exceeded), it writes its state to the completion queue, which the scheduler relays to a so-called context stream.
3.3 Spiders

A spider is a Domino add-in process which will invoke threads to explore different repositories. Different spider types have been provided, designed to extract content from various content repository types.
Once a spider process is started, it can start any number of additional threads to explore different repositories, including Notes, QuickPlace, Domino.Doc, e-mail (Domino), Web and file system. This means that one LDS can have a Notes spider and a second LDS can have a Notes and file system spider.

The queue of repositories that are scheduled to be spidereed or re-spidered for changes is populated by the Discovery Server Control Center scheduler, which constructs each request from the data repository forms in dashboard.nsf. The scheduler collects the information from the Data/Repository (which is specific and different for each repository) and Data/Spider Settings section (which is general and the same for all spiders of that type).

A spider polls discoveryserver.nsf (which is named “Discovery Server Control Center” and stores meta data about the repositories to be spidered) to determine the number of concurrent spider threads, whether it is enabled, and when it is scheduled to be active. When a spider is active, it checks the work queue for repository to process.

Processing is spider-specific, and each spider has different techniques to traverse and find all the content in a repository. The data repository spiders read data from file systems, Web sites and Domino databases to write the normalized data to the metrics DB2 tables.

Each entity (i.e. document, file, or URL) is processed and registered with the K-map (stored physically in DB2 tables). Relevant information is converted into XML and written to three queues, which are processed by ATG, Metrics and FullText.

The XML contains either control information (i.e. Start and Stop messages for a repository, issued to the Completion queue, handled by the scheduler), or information from each document processed. Besides this output, the spiders also write status information to the Completion queue. Messages in the Completion queue are processed by the scheduler, which controls when a repository should be spidereed next; these messages are also displayed in the Lotus Discovery Control Center (discoveryadmin.nsf).

3.3.1 Where do the spiders store the spidereed information

The spiders store data repository content and organizational information from the repository as normalized XML to four queues, as follows:

**XML queues**

- TaxonomyQueue, where Taxonomy context information and document information are stored
- FulltextQueue, where fulltext context information and document information is stored
- MetricsQueue, where server transactions, metrics context information, document informations and database transaction information are stored
- Completion Queue, used by the scheduler for updating repository information (Start/Stop information and information for requeueing and cleanup purposes)

When scheduled to run, the queues are processed by the corresponding worker tasks and the scheduler. They read the information from the queue and either write it to the databases, or initiate the necessary actions on the corresponding spider tasks (see Figure 3-3).
3.3.2 Webspider - basics

The webspider is the most complicated of the three spider types, because it has to traverse a Web site by links (anchors), and not by directory structure. This makes the occurrence of previously spidered content in an unspidered link tree possible, since the Internet or a Web site has no defined start or end.

To avoid redundant spidering of already traversed URLs, the webspider maintains a so-called URL table, which contains all URLs spidered in this session by this particular webspider before.

**Note:** In the first release, the webspider does not write this list to disk, but instead keeps it in memory for the time of the current traversal. Therefore, depending on the size of the Web site you have defined as a data resource, this list might consume large amounts of RAM and virtual memory while the spider is running.

The current recommendation for webspider settings is: do not follow links to other servers when the level to be spidered is set to a maximum of 5.

The webspider is started and stopped by the Discovery Server’s scheduler and given a certain time window by the user-defined spider schedule during which the spiders will process their private work queue.

**Note:** The webspider can also be stopped manually by typing: `tell webspider quit` to the LDS console; this will stop the spider, saving its status. Typing: `load webspider` will load the webspider.

When the webspider recognizes an item in the webspider work queue, it removes it and goes through the following process.

**Webspider control instance**

1. Check if the URL already exists in the URL table:
   a. If not, add URL to URL Table; all other values are set to NULL.
   b. If it exists, continue.
2. Use a unique private work queue. Check if it already exists and if the spider is being restarted:
   a. If it is a normal restart message, then a queue must exist and it does not need to initialize the `<set>` which was stored in the context.
   b. If it is an error restart message, then we have lost the `<set>` information and we just begin processing the URLs in the queue.
   c. If we are not being restarted, then the queue should be empty.
3. Create a <set> and pass this to child threads. The <set> is used to prevent redundant traversals of URLs.
4. Process the include and exclude list and create internal regular expressions data structures.
5. Create <n> child threads (<n> is defined by the user in “Discovery Control Center”).
6. Add the starting seed URL to the private work queue. In addition to the URL, it contains the following item: remaining levels to traverse.
7. Wait until the queue is empty or it is time to yield (outside schedule window) and the child threads are idle:
   a. Write progress information to the Completion queue.
8. Write a Completion message to the Completion queue:
   a. If we’re done, write a Completion message.
   b. If we’re leaving because we need to yield, then write a restart message which contains all the data from the <set>.

**Webspider child threads:**
1. Check to see if the schedule window is still valid. If so, continue. If not wait in an idle state.
2. Read a task from the private queue.
3. If empty, wait and go back to step 1.
4. Set state to busy.
5. Get the URL and check whether the URL has changed since it was last processed. To check whether a URL is new (usually the document this URL represents), the webspider retrieves the Last-Modified field from the Web server and compares this value with the one stored in the metrics. If it is new, then we always need to process. If it is already in the DB2 URL table, then check the date. If the server does not support the modified date, then fall back to size and/or other techniques.
   a. If it has changed, convert to XML and dispatch to Full Text, Taxonomy, Metrics, maybe 1.
6. Parse the URL for links and perform the following for each link:
   a. Check whether this link should be added to the queue: Check robots.txt (according to the robot exclusion standard [Mauldin et al. 1996] Spidering BOF Report, (http://www.w3.org/Search/9605-Indexing-Workshop/index.html )).
i. Cache the last robots.txt processed, and if working on the same server, do not fetch and parse it again.

7. Check whether this URL is included in the <set>. If it is, then it has already been traversed. If not, add it to the set.

8. Check that remaining levels to traverse is not zero.

9. Add a new task to the private work queue. It contains the following items besides the URL: remaining levels to traverse. Remaining levels to traverse is calculated by subtracting one from the level passed in.

10. Set state to idle.

**Note:** Since the webspider is reading and writing the URL table from the K-map (stored in DB2), this has to be considered when distributing services between LD servers.

When spidering Web sites, it is important to check the structure and types of HTML used on the Web site first. If the content is presented using, for example, Java applets, the webspider will not be able to correctly read the presented documents at all. To have the same rich amount of meta data spidered, the webspider needs the html documents to provide certain html elements and so-called meta tags.

For more information about the Last-Modified date, refer to RFC2616 at:

http://www.ietf.org/rfc/rfc2616.txt?number=2616

The webspider supports (parses) these HTML elements and normalized to XML:

- Meta
- A (Anchor)
- Text
- Frameset
- Title

It also supports these meta tags:

- codepage
- title
- subject
- author
- keywords
- comments
- category
- notes
- manager
- company
You can therefore map meta tags to Discovery Server entities, in order to have webspidered contents added to the taxonomy with a rich set of meta-information, just like you can have in a Domino database.

If you have problems spidering very large Web sites, you may want to change the detail level of the webspider log file. You can change the way errors and warnings are logged by adding the following line to the notes.ini file:

- KDS_WEBSPIDER_WARN_LEVEL=<level of logging in values 1, 2, or 3>

where:
- 1=log nothing
- 2=filter the logged messages
- 3=log everything

**Note:** Setting KDS_WEBSPIDER_WARN_LEVEL=3 may result in very large webspider logs.

### 3.3.3 File system spider - basics

The file system spider is a very useful tool to use for spidering even completely unstructured file repositories with large amounts of different files using different file types—while still respecting the security settings this file system may provide.

The file system spider is passed a networked file resource and recursively works the directory structure on the path. Just like the webspider, the file system spider keeps track of previously spidered files, because the files could have been moved, copied, or altered. To avoid redundant re-spidering of these files, time stamps, modified dates and so on are processed and monitored in a private file system spider list.

The file system spider can process all file types supported for conversion by the Verity KeyView file viewer, also used in Domino R5, which are (among others):

- Microsoft Word for Macintosh
- Microsoft Word
- Rich Text Format
- Microsoft Word
- Filemaker MACLotus PICLotus Ami Pro Draw
- WordPerfect
- Lotus 1-2-3
- Microsoft Excel
To see a complete list of supported file formats, refer to Appendix A, “File types you can spider” on page 249.

Note: When spidering file systems, keep in mind that the file system spider is very resource-consuming (especially network bandwidth), because it opens every file repeatedly for the initial spider run of the given data repository.

3.3.4 Notes spider

The Notes spider’s job is similar to those of the file system spider and the Web spider. It gets a list of new or altered documents in a Notes database and spiders those documents. If the database is a new data repository, it processes all documents.

Time stamps are compared to check whether a document has been altered since the last Notes spider run. The list of documents to be spidered gets passed to a conversion function, which converts all items to XML. This normalized XML "stream" is passed to the metrics queue, K-map indexing, and taxonomy queues.

Similar to the Notes spider, there is also an e-mail spider designed specifically for spidering Domino e-mail databases.

3.3.5 Profile source spider

The profile source spider is used to extract people information from directory resources. The repository types used for this extraction process are either Domino Directories or LDAP Directories.
Similar to all other spiders, it writes data drawn from a repository to an XML stream, which is then mined for information by a metrics add-in task. The profile source spider is able to spider various repositories and have them mined into a single people.nsf profile representation, aggregating the information.

When spidering an LDAP Directory, the profile source spider spiders the meta data in the following fields—which are also retrieved by the Lotus Discovery Server from an LDAP Directory server to authenticate a user:

- AltFullName
- Certificate
- Firstname
- Fullname
- HTTP_Hostname (Server)
- InternetAddress
- Lastname
- Listname
- Location
- MailAddress
- MailDomain
- MailFile (Person)
- MailServer (Person)
- Members
- PublicKey
- ShortName
- userCertificate

### 3.3.6 Queue handling

The most important control files for all services in the Discovery Server System are the .queue files. They control which data repositories are in what state of process, hold information to be passed on for further processing, and even enable load balancing between a large number of threads working the same data repositories.

The Discovery Server uses the following queue files:

- ATGOutOfBandClassifyQ. Queue written by ATG. Read by ATG. Contains documents to be moved to the “uncategorized” cluster.
- CompletionQ. Queue written by any consumer task. Read by the scheduler. Contains Start/Stop messages.
- LDSATGCmapWorkQ. Queue written by all CMap “workers”. Read by ATG. Contains change messages from CMap to ATG
- LDSMetricsWorkQ. Queue written by the spider. Read by MetricsWorker. Contains XML stream.
- LDSProfileSyncWorkQ. Queue written by the spider. Read by MetricsDirSyncWorker. Contains XML stream.
- LDSTaxonomyWorkQ. Queue written by spider. Read by ATG. Contains XML stream.
- FileSysSpiderQ. Queue written by the scheduler. Read by the spider. Contains repositories to the spider.
- NotesSpiderQ. Queue written by the scheduler. Read by the spider. Contains repositories to the spider.
- ProfileSpiderQ. Queue written by the scheduler. Read by the spider. Contains repositories to the spider.
- WebSpiderQ. Queue written by the scheduler. Read by the spider. Contains repositories to the spider.
- XMLSpiderQ. Queue written by the scheduler. Read by the spider. Contains repositories to the spider.

In the example of the metrics data flow diagram (see Figure 3-3 on page 42), the person information found in an LDAP or Domino directory is parsed to XML by the MetricsDirectorySync thread (which is, in fact, the DirectorySyncSpider). This XML data is temporarily stored in the DirectorySyncWorkQ, then read again by the MetricsDirectorySyncWorker, creating the people documents to the people.nsf database. So in this context, the queue is used as a sort of temporary storage, to make the asynchronous and (even more important when thinking about scalability) concurrent processing of people directory information possible.

In case of the webspider, it holds information about which seed URLs (data repositories) are on schedule, what is the maximum number of levels to be traversed, which user ID and password to use for URLs, and other preferences.

In addition to the Notes spider and file system spider, every webspider uses its own work queue (private .queue file, made unique by concatenating the REFERENT ID to a standard name), storing a set of URLs already traversed to prevent the redundant traversing of them.
3.4 Which tasks belong to which Lotus Discovery Server component

The Lotus Discovery Server components mentioned in this chapter all have a representation in the actual Domino server console. Following are the names of the executables and the component they represent:

- natg.exe (atg) - represents the ATG control instance
- nkmapbuild.exe (atg) - K-map building
- nkmapserve.exe (KDS aka Content Map) - Discovery data store
- ndssched.exe (DS_sched) - represents the scheduler
- nmetrics.exe (metrics) - represents the Metrics control instance
- nnotesspider.exe (NotesSpider) - represents the Notes spider control instance
- nwebspider.exe (WebSpider) - represents the webspider control instance
- nfilesysspider.exe (FileSystemSpider) - represents the file system spider control instance
- nportalsetup.exe (Portal Setup) - checks for correct settings of the Lotus Discovery Server
- nkmapindex.exe (KMapIndex) - full-text indexes the K-map

3.5 Metrics

The Metrics task is a very complex, multithreaded server task, running various components as “threads”.

Figure 3-3 on page 42 will help you better understand the following components and their interaction with queues, spiders, and the scheduler; it shows the DirectorySync spider (named Metrics Directory Sync in the diagram) and its interaction with the metrics.

3.5.1 Metrics processing (calculation, updating, and reporting)

As you can see in the figure, the Metrics add-in task consists of various workers and one spider. To make this architecture more understandable, Metrics is logically divided in two parts: Metrics Processing and Metrics Collection.

Metrics Processing consists of the two following tasks:
AffinityWorkerTask, AffinityCalcTask

The AffinityCalc task, which is a thread in the MetricsAddIn task, calculates the affinity (a value describing a person's interest in a certain topic or K-map category) a person has to a certain topic. The calculation of affinities is an ongoing process, always calculating the new affinity value based on the affinity value this individual person had on this particular topic before.

This affinity value gets passed on to the MetricsAffinityCalc task, which (if set to do so in the LDS Administration interface) initiates the mail workflow with the user (to let the user accept or decline the proposed affinity).

If the affinity proposal is accepted by the user, the affinity value gets published to the people profile database (people.nsf) and stored in the corresponding DB2 tables of the metrics database.

Note: In addition to the ability to decline proposed affinities, users can approve or decline having their mailfiles spidered to determine any relationships between Domino mail content and K-map category areas, to strengthen affinity calculations of profiled users.

This is a necessary control (for end users, in particular) because some countries require that end users be notified that data is being collected electronically and approve the use of that collected information. For more information about data privacy, consult:

http://www.ibm.com/privacy

3.5.2 Collection

MetricsCollection is actually a name for the MetricsWorker (a thread of the MetricsAddIn task) that accesses the MetricsWorkQ to read a stream of XML documents from it and mine each of these XML documents for certain raw metrics data. When the XML of each document has been mined, the metrics data is written and updated in the metrics database (tables in DB2).

The raw XML documents in the stream extracted contain entities for each of the fields, defined in the mapping documents for your spidered repositories (e.g. $Global map).

Document scoring/value

Document scoring is an approach to evaluate the content of a document in the K-map. This value is controlled by the following “triggers”, which you can re-sort by means of their weight on the computed document value:

1. Links to a document.
2. Links from a document.
3. Responses to a document.
4. Times a document has been opened using the K-map.
5. Recency of the last update to a document.

The top-most trigger in this sorted list represents the value with the highest weight on the calculation of the document value.

The higher this number “value” is, the more useful the associated document is meant to be for the users. You can change the order of these triggers in the metrics settings.

The “document fit value” (which may be viewed using the K-map editor tool) is a computed number representing the fit of a particular document into a certain category obtained through the K-map building clustering process.

![Figure 3-4 Document cluster in 3D vector space](image)

This value refers to a similar attribute computed for the K-map, telling you whether a document fits into a certain category perfectly, or doesn’t fit completely in a category but—in terms of content relation—fits this particular category rather than another. This value could also be referred to as a “vector distance”, a vector in the k-mapdocument space (representing a document) from the very center of a K-mapcluster, which is, in fact, a category in the K-map.
3.6 K-map

The following sections explain how K-map Building Services (sometimes referred to as ATG) and its automatic document clustering technology work.

3.6.1 Stopword list

Every LDS installation leverages a set of language-specific, predefined stopwords.txt files, the stopword list. These files contain words that will not be used to cluster documents from. The stopword files may be edited by organizations wishing to identify specific words that may occur frequently from being included in the K-map building content evaluation processing. The words contained there will be removed from the XML content stream that the ATG task uses to build the taxonomy. The words in the stopwords.txt files are case sensitive.

The stopword list is unique to every language you may want use, and can be edited with a standard text processing utility (for example, MS Notepad).
3.6.2 K-map building

In order to build the initial Knowledge Map for the Lotus Discovery Server, the LDS Scan Task continually scans the LDSTaxonomyWorkQ.queue file for new XML content being written by one of the three LDS spider types; refer to Figure 3-6 on page 55.

Important: When editing the stopword file, do not enter the same word in the list twice. (This could happen, for example, with words that have different meanings but are spelled the same way if you forget that you already added the word to the list.)

To check for this problem, sort the list alphabetically to make it easier to find already added words.
The following is a description of what happens during the scan:

1. The Scan Task compares an XML document against all stopwords lists on the LDServer. Words found in the XML stream and in the stopwords files are removed from the stream.

2. InXight Tools Tokenizer marks the remaining words as tokens, and spaces between words (tokens) are removed.
3. **InXight Tools Stemmer** reduces the amount of data by trying to reduce words to their stem (e.g. tokens => token, sleeping => sleep etc.).

4. **InXight Tools Tagger** eliminates language-specific parts of speech.

Following are the files that are written by the InXight tools in the ongoing process of K-mapbuilding:

- **InXight_cnt** - Occurrence and count of tokens per document (records document within cluster)
- **Label_InXight_tok** - Occurrence and count of tokens (for labels, nouns only) per document within K-mapcluster
- **InXight_tok** - containing individual tokens and the number of occurrences within a K-mapcluster
- **Label_InXight_cnt** - Individual tokens (nouns only are used for category labels) and their count by K-mapdocument cluster
  - **Tokenizer**: Evaluate metadata for occurrence of tokens (are words; all non-white space). Filter items such as punctuation, numbers (20,40), non-alphanumeric words (“/home/q”).
  - **Stemmer**: Reduce tokens: Nouns, verbs, adjectives to their root form (for example: running to run) using language-specific processors.
  - **Tagger**: Identifies the part of speech (verbs, adverbs) for each token and also identifies noun tokens which are used later to determine category labels.
- **The results, with additional analysis by Inxight tools as outlined below, are written to specific Discovery Server files utilized later during K-map Building processes:**
  - **InXight_tok** - List of unique, parsed tokens. This file is added to as new content is evaluated through the K-map Building processes.
  - **InXight_cnt** - List of tokens and their count per document.
  - **Label_InXight_tok** - List of unique, parsed nouns (label candidates).
  - **Label_InXight_cnt** - List of noun tokens and their counts per document.

5. A token is also a document's weight. Tokens occurring in title fields are counted 3 times, tokens occurring in subject fields are counted 3 times, and tokens occurring in KeyFields are counted 2 times.

6. Every token is assigned a unique ID.

The ATGRoot DocID file associates individual document IDs with the number of occurrences of content and label tokens per document.
The ATGRoot DocInfo file maintains state information about each document processed: whether it is included in the initial training set, if it has been categorized yet, or read only by the scan process.

7. XML documents having 30 or less tokens are forwarded to the classification queue (ATGOutOfBandClassifyQ.queue), not being processed in the initial K-map clustering process.

8. If the number of documents exceeds a certain threshold (defined by the Administrator in the Discovery Server Administration database), K-map building is started.

The K-map building (clustering) is done by the IBM SABIO Tools for document clustering:


The technique of vector-categorization is used to place all documents (vectors) into an n-dimesional vector space, trying to find clusters (categories) these documents could fit to. These clusters can overlap each other, or consist of a number of smaller clusters (subcategories).

**Standard procedure for automatic document clustering**

1. All words in the (XML) documents are counted.

2. Certain words are selected; these are called “cluster terms”.
   a. This selection is based on a value called “term-discrimination value”, which is used to balance the size of all clusters, making them not too big and not too small.
   b. Having selected the cluster terms ensures that the vector space is not too wide or too narrow to find appropriate clusters. This has to be done to put the clusters in one single cluster tree, called the taxonomy. The cluster terms represent dimensions in a multi-dimensional space (each cluster term represents one dimension), to be used in placing documents in this vector space.

3. Documents are placed in the vector space. The more frequent a cluster term is counted, the higher the value of the document in the dimension of that cluster term. This is done for every cluster term (dimension) until the document can be placed into the vector space as a point or vector (from the origin of the vector space to the location in the vector space where a document has been calculated to be placed).

4. Documents or vectors that are near each other are defined to belong to one cluster.

5. This cluster is named after the cluster terms found in most documents in this cluster.
a. Cluster terms found in fewer documents are not selected to label a cluster (even though the documents containing this particular cluster term still belong to this cluster).

6. Clusters that are found near each other in this multi-dimensional vector-space are clustered into one cluster, being named using the same method as with documents. This is done until there is only one cluster left over, representing the root of the taxonomy tree.

This is a general overview of how automatic document clustering works.

Note: As previously mentioned, SABIO (ATG) uses a very similar method to cluster documents, using the term “tokens” instead of “words”.

3.6.3 K-map editing

The K-map editor is a standalone application which can be installed on any Windows client machine in your network; just have your taxonomy editors click the following URL and follow the instructions to install the editor on their local machine:

http://primaryserver.domain.com/kmapeditorinstall.html

The user ID that is used to authenticate against the Primary LDS Server should be the flat user name-type id connecting to the domain name server name of your primary LDS server (for example, server.domain.com). It should also be a member of the TaxonomyEditors group (you may want to add your KDS Administrators group to them).

The K-map Editor application establishes direct communication with the K-map, rather than going through a work queue. When you start the editor, you may occasionally see the message: The K-map Building service is currently updating the category tree.... This message indicates that the Editor is trying to get exclusive access to the K-map so the server isn't changing the K-map while you're trying to edit it.

This locking mechanism does not lock out other K-map editors. When you start the K-map Editor and connect to the K-map, the server stops the K-map Building task (in essence, stopping the server from modifying the K-map while you're editing). The server will automatically resume managing the K-map once you close your editing session.

The K-map must keep ATG, Metrics and FullText informed whenever any changes are applied to it. Some operations performed by the taxonomy editor need to be communicated quickly to ATG for the following reasons:
They affect all pending messages in the normal ATG workqueue. For example, if the ATG normal queue has 1000 messages and an editor modifies the taxonomy, you want the editor’s modification to happen before the 1000 pending messages are processed. If the editor’s request happens later, then some of the 1000 messages might need to be reprocessed again.

Some operations (such as “sub divide” and categorizing uncategorized document) need to occur as soon as possible, because an editor is waiting to review the result. We understand that these actions are not immediate, but since an editor is waiting, they should occur before other processing.

- This will be accomplished by the CMap and ATG using a separate work queue. ATG will always check this work queue before reading a message from the normal work queue.

By communicating directly with the K-map, all edits you make happen immediately. If you went through a work queue, you wouldn’t see any changes until the server processed the requests in that queue, which would make it hard to track the changes you make during that editing session.

### 3.6.4 K-map searching

Search queries submitted to the K-map information retrieval system are done using full text queries, while the full text index covers categories, documents and full text indexed databases (especially the people.nsf database). In the case of the K-map of LDS, since the Content Map (CMap) stores all the documents the taxonomy consists of, this is the fastest way to search for content.

![Figure 3-7 Query and document vector representation in a three-dimensional space](image.png)
To make the K-map browsing UI faster, there is a caching mechanism with a configurable expiry time, caching all the information that has been queried from the DB2 backend databases. (This does not include the K-mapsearch UI or the KMapEditor.)

Per default settings, this cache is refreshed every 60 minutes, and is populated when the first query on the K-map is submitted by any user. The notes.ini settings to change the presets for the K-map cache are:

- KDS_CURSOR_MEMORY=<maximum amount of data to cache in memory in megabytes; the default is 10 megabytes>
- KDS_CURSOR_DISK=<maximum amount of data to spill to disk in megabytes; the default is 100 megabytes>
- KDS_CURSOR_TIME=<time in minutes to hold data in the cache; the default is 60 minutes>

When the cache reaches a size of 110 MB (when the default setting of 100 MB cache size is used), the least recently used cache entry is expired. When KDS_CURSOR_TIME in minutes has passed since the last cache update, the cache is flushed.

**Note:** All documents ACLs are also cached.

When searching for content, the K-map uses the back-end features listed in Table 3-1:

<table>
<thead>
<tr>
<th>Table 3-1 Which search type uses which feature</th>
</tr>
</thead>
<tbody>
<tr>
<td>Search type</td>
</tr>
<tr>
<td>---------------</td>
</tr>
</tbody>
</table>
| Categories about... | ▶ phrase matching (match entire string exactly, even if unquoted)  
▶ fuzzy search (match alternate spellings)  
▶ partial-word matching (match word containing search term)  
▶ stemming*  
▶ thesaurus* |
| Documents about... | ▶ phrase matching  
▶ fuzzy search  
▶ partial-word matching  
▶ stemming*  
▶ thesaurus* |
To search in the K-map, users can use the operators listed in Table 3-2:

<table>
<thead>
<tr>
<th>Search type</th>
<th>Feature used</th>
</tr>
</thead>
<tbody>
<tr>
<td>Documents authored by...</td>
<td>▶ people name aliases</td>
</tr>
<tr>
<td>People named...</td>
<td>▶ people name aliases</td>
</tr>
<tr>
<td>People who know about...</td>
<td>▶ phrase matching</td>
</tr>
<tr>
<td></td>
<td>▶ fuzzy search</td>
</tr>
<tr>
<td></td>
<td>▶ partial-word matching</td>
</tr>
<tr>
<td></td>
<td>▶ stemming*</td>
</tr>
<tr>
<td></td>
<td>▶ thesaurus*</td>
</tr>
<tr>
<td>People whose profile contains...</td>
<td>▶ phrase matching</td>
</tr>
<tr>
<td></td>
<td>▶ fuzzy search</td>
</tr>
<tr>
<td></td>
<td>▶ partial-word matching</td>
</tr>
<tr>
<td></td>
<td>▶ stemming*</td>
</tr>
<tr>
<td></td>
<td>▶ thesaurus*</td>
</tr>
<tr>
<td>Places about...</td>
<td>▶ phrase matching</td>
</tr>
<tr>
<td></td>
<td>▶ fuzzy search</td>
</tr>
<tr>
<td></td>
<td>▶ partial-word matching</td>
</tr>
<tr>
<td></td>
<td>▶ stemming*</td>
</tr>
<tr>
<td></td>
<td>▶ thesaurus*</td>
</tr>
</tbody>
</table>

* Only for languages supported by stemmer (InXight)/thesaurus (files can be modified).

Table 3-2  Valid K-map search operators

<table>
<thead>
<tr>
<th>Operator/Expression</th>
<th>Example</th>
<th>Meaning</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACCRUE</td>
<td><em>,,</em></td>
<td>Cat, Dog, Fish</td>
<td>Weighted OR</td>
</tr>
<tr>
<td>AND</td>
<td>&quot;&amp;&quot;</td>
<td>Cat AND Dog</td>
<td>Boolean AND</td>
</tr>
<tr>
<td>OR</td>
<td>&quot;</td>
<td>Cat OR Dog</td>
<td>Boolean OR</td>
</tr>
<tr>
<td>NOT, &quot;!&quot;</td>
<td>Cat AND NOT Dog</td>
<td>Boolean NOT (Exclusion)</td>
<td>Need AND/OR before NOT</td>
</tr>
<tr>
<td>&quot;-&quot;</td>
<td>Cat -Dog</td>
<td>Exclusion</td>
<td></td>
</tr>
<tr>
<td>&quot;+&quot;</td>
<td>+CAT +DOG</td>
<td>Requirement</td>
<td></td>
</tr>
<tr>
<td>Operator/Expression</td>
<td>Example</td>
<td>Meaning</td>
<td>Comments</td>
</tr>
<tr>
<td>---------------------</td>
<td>---------</td>
<td>---------</td>
<td>----------</td>
</tr>
<tr>
<td>PARAGRAPH</td>
<td>Cat PARAGRAPH Dog</td>
<td>Paragraph Proximity</td>
<td></td>
</tr>
<tr>
<td>SENTENCE</td>
<td>Cat SENTENCE Dog</td>
<td>Sentence Proximity</td>
<td></td>
</tr>
<tr>
<td>TERMWEIGHT</td>
<td>TERMWEIGHT 80 Cat AND TERMWEIGHT 30 Dog</td>
<td>Weighted Search</td>
<td>Need AND/OR before subsequent TERMWEIGHTs</td>
</tr>
<tr>
<td>FUZZY</td>
<td>FUZZY 66 Cat OR FUZZY 80 Dog</td>
<td>Weighted Fuzzy Search</td>
<td>Need AND/OR before subsequent FUZZYs</td>
</tr>
<tr>
<td>NEAR</td>
<td>Cat NEAR Dog</td>
<td>Proximity</td>
<td></td>
</tr>
<tr>
<td>EXACTCASE</td>
<td>EXACTCASE &quot;dog&quot;</td>
<td>case sensitive search</td>
<td></td>
</tr>
<tr>
<td>&quot;*&quot;</td>
<td>&quot;?&quot;</td>
<td>Dog* OR Cat?</td>
<td>Wildcard</td>
</tr>
<tr>
<td>&quot;&quot;</td>
<td>&quot;This is a short phrase&quot;</td>
<td>Text phrase search</td>
<td></td>
</tr>
<tr>
<td>ACCRUE with Proximity</td>
<td>Cat ACCRUE Dog Near Water</td>
<td>Find documents that contain both &quot;Cat and Dog, then find Water near Dog</td>
<td></td>
</tr>
<tr>
<td>ANDs with ORs and NOTs</td>
<td>Cat AND ! Dog OR Fish</td>
<td>Find documents that contain &quot;Cat&quot; but not &quot;Dog&quot; or contain &quot;Fish&quot;</td>
<td></td>
</tr>
<tr>
<td>Weighted search and ORs</td>
<td>TERMWEIGHT 20 Dog OR TERMWEIGHT &quot;Fish&quot; OR TERMWEIGHT 80 Cat</td>
<td>Find documents with either &quot;Dog&quot; or &quot;Fish&quot; or &quot;Cat&quot;, rank them by the sum of their termweights</td>
<td></td>
</tr>
</tbody>
</table>
K-map building recommendations (initial/customizing)

In general, it is best to use fairly small data repositories, focusing on particular topics, with medium to large RichText (body text) content per document.

Start identifying topics your users work on first and try to find “reference experts” who know about the topic and the data source you are about to use. It is recommended that you start with databases or document repositories of no more than 200 to 300 documents per repository which you or your experts know pretty well.

Do not use repositories on too many topics when initially creating the taxonomy, and try to avoid using Web sites as initial data repositories, especially if you do not know for sure how many documents there are on the Web site. It is best to add Web sites/documents to an already created taxonomy.

Start building the initial K-map with a training set of documents in a notes database. Try to identify documents that best represent your business and the topics you need to know about. You might also create a Domino database and paste content in there from different sources you need to have categories on in your taxonomy.

Keep in mind that the fewer categories you create in your initial taxonomy, the easier it will be for your taxonomy editors to rename category labels, re-sort documents from one category to another, and create new, better-fitting categories for documents in a particular (automatically created) category. Remember: the categorizer learns how you want your taxonomy to look—and will make taxonomy editing more and more unnecessary—the more you edit your taxonomy.

When customizing an already created taxonomy, talk to your users and the people that will use the taxonomy. Ask them if they like the way documents are sorted into categories, or which categories they think they need.

<table>
<thead>
<tr>
<th>Operator/Expression</th>
<th>Example</th>
<th>Meaning</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>ORs and NOTs</td>
<td>Dog OR ! Fish</td>
<td>Find documents that have &quot;Dog&quot; or don't have &quot;Fish&quot;, rank those highest for which both conditions are true</td>
<td></td>
</tr>
</tbody>
</table>
3.6.5 Field maps

To correctly retrieve meta data information from any document source (for example, author information, creation date and so on), LDS uses “field maps” to be able to adopt to any data source you might have and to be able to convert the correct content to the XML representation which is stored in the K-map.

There is always a $Global field map and a new field map that may apply to one or more data repositories you define.

The $Global field map is always used, but it can be altered. If you have to edit the $Global field map, be aware that mapping non-existent fields of your data repository to the special LDS fields of an entity will result in incorrect or poor meta data information for all repositories.

To map non-standard field names of repositories, you might want to spider to the LDS field names. You should consider creating a new field map, which will then be used in addition to the $Global field map.

If you do not create a mapping for a field in this new map, the LDS service will use the $Global for that entity of your data repository. Field names you alter in the new field map overrule the $Global field map.
Figure 3-8  Data repository settings with additional field map specified

When you edit a new field map for a certain repository, be sure to check the correct field names in the data repository you want to spider before you schedule and save the repository preferences.

You can enter multiple field names to map to one LDS Field name entity by separating them with commas.

**Note:** The box “Don’t map” was not functional in KDS 1.0.
3.6.6  K-map indexing

In order to find all documents clustered within the created taxonomy, LDS utilizes the K-map indexer, indexing all XML documents put onto the LDSFullTextWorkQ.queue.
When a document is indexed by LDS, it indexes all contents of the documents (stopwords are not indexed). The index data is broken out by repository into different indices. The LDS Fulltextindex server has one logical index, which is made up of 8 separate physical index segments. This number is chosen, because it allows faster indexing.

The index files live in the file system under \Lotus\KDS\Data\ftdomain.di in 8 separate directories named LDSIDX00 through LDSIDX07. Person Profile documents are indexed into a separate Fulltextindex, which also resides on the Fulltextindexing Server.

3.7 Profile synchronization

The main purpose of the Profile Synchronizer is to track changes to the organizational and contact information for a certain person.
The first part of profile synchronization is mainly processed by the Metrics Affinity Task (MetricsAffinityCalc-Task and Metrics Affinity Worker).

Data in the Metrics database is processed by the Metrics Affinity Calc Task, calculating the affinity values between people and categories. Discovered affinities are then proposed to the Metrics Affinity Worker, which sends out an e-mail with the proposal to the corresponding users. If a person accepts an affinity proposal, that affinity is published by the Metrics Affinity Worker to the people directory database.

The second part, updates or creation of people profile documents, is done by the Metrics Directory Sync Worker Task, reading update requests from the Directory Sync Work Queue and publishing or updating people documents in the people directory database.

### 3.8 Affinities

Affinities are, as already stated, calculated and published by the Metrics Affinity Calc Task and the Metrics Affinity Worker task.
To get a published affinity to a certain topic, a person has to demonstrate a strength of relationship to existing K-map category areas via their interactions with content:

- Author
- Edit
- Link to
- Respond to Documents
- Read documents, using the KMap

**Note:** The default threshold is set to “100” which should be reconfigured by the administrator after initial setup to a lower value to calculate affinity strengths for users. If the threshold remains at 100%, no users will match that criteria and thus you will have very few (or more likely, no) affinities discovered.

If a person “acts” on documents more often than <threshold> (where <threshold> is a number <= 100) percent, the metrics affinity calc task proposes an affinity between the person and the categories where the documents were found that triggered the affinity calculation task to discover an affinity.

Translated into a real-world use case, if a person is discovered to have an affinity threshold of 85, for example, then this person has worked more on documents of a certain category than 85% of all LDS users. At that point, an e-mail is sent to the profiled user asking her to approve or disapprove having that category term published to their profile as an affinity.

### 3.9 Categorizer

Once the initial set of clusters has been created, K-map Building service (ATG) provides an automatic classifier. Through this service, the K-map Building process compares the words in new documents (and documents in the “Uncategorized” section of the K-map) to the words in the clusters it has already created.

If the new documents are similar to the documents already in existing categories, the new documents will appear in the same categories. If the new documents are not similar (e.g., do not use similar cluster terms as the documents already clustered into categories), ATG puts them in an “Uncategorized Documents” category for evaluation by the human editor in the process of K-map refinement.
ATG also creates new subcategories when the maximum number of documents per category level exceeds a certain value. (This value can be set in the “Lotus Discovery Control Center”, “Startup” Navigator, “3. Create K-map”, “Begin K-map creation”, “Step 2. K-map settings”.)

They are labeled using the same technique when creating a initial taxonomy.

You enable categorization by clicking **Begin Categorizing** in the Discovery Server Administration interface.

**Note:** Do not turn on categorization until you are totally sure your K-map editors have finished the initial relabelling process.

### 3.10 Multilanguage support

The components of LDS which have to work on repositories with different languages are:

- ATG Scan Task
- InXightTools Stemmer
- InXightTools Tagger
- K-mapSearch Feature

The ATGScan Task removes words from the XML document stream which can be found in one of the stopwords.txt files, one file for each language supported. When editing the stopwords files, ensure that there are absolutely no duplicate words in the list; otherwise, this might crash the scan task.

**Note:** You might want to add words here that you don’t want the classifier to create categories (clusters) with.

The InxightTools Stemmer reduces the words found in the XML document stream to their stem (for example, reducing “driving” to “drive” and “cars” to “car”). The InxightTools Tagger eliminates language-specific parts of speech.

The InxightTools require a set of special files for every language to be supported, which is deployed with the installation LDS. These files usually do not need to be customized and are ready to be used without editing.
To have the search feature work perfectly, you need to provide a thesaurus that is customized to fit the requirements of your business and of your language in general.

To fully support search for accented characters (ö,ü,o,â,ñ etc.), enable the following notes.ini parameter:

FT_INDEX_ACCENT_SENS=1

### 3.11 Thesaurus

KDS 1.0 shipped with a thesaurus tool called Lotus Discovery Server Thesaurus Tool, which is based on the IBM LanguageWare thesaurus. A detailed description of how to use the thesaurus is available in the readme.chm file of LDS 1.0.
Setting up a successful test environment

When dealing with complex software such as the Lotus Discovery Server provides, it helps if you understand the main elements and functions of the software and how they all work and fit together. This requires many years of experience working with the product, or if you are new to the product, learning about it as quickly as possible. We find that the best way to gain experience is through hands-on practice—setting up a test environment where you can see the functionality of Lotus Discovery Server, without worrying about impacting your corporate production server environment.

In this chapter we show you the best way of setting up an environment for testing purposes. We give step-by-step instructions that enable you to install Lotus Discovery Server, even if you are not very familiar with Notes/Domino.
4.1 The scope of this chapter

In this chapter, we show how to set up Lotus Discovery Server in your company, whether as a KDS server in an existing Domino domain, or registering Lotus Discovery Server in a new Domino domain. We focus more on the second scenario since it is more likely that, for a test installation, you will install and configure the KDS server in a new Domino domain.

Note that before doing a rollout to the enterprise, the following steps must be completed. We provide detailed descriptions for each step.

1. Set up and configure the first Domino server
2. Set up the administration client
3. Register Sametime and KDS server and users
4. Make changes to the server document to suit the requirements of KDS
5. Create group documents and connection documents
6. Configure the Sametime server “on top” of the Domino server
7. Configure the KDS server
8. Prepare the data to be spidered

We also provide hardware and software configuration guidelines, and discuss the following:

- Suggested load sequence on the KDS server
- Which services you need to enable and when
- Metrics considerations for the test environment
- Safe reset points on the KDS when running a backup

4.2 Server setup

A typical test server setup is shown in Figure 4-1 on page 75. As you can see, there is a Domino server which typically stores a directory of people and may also contain many databases of information, a Sametime server used for instant messaging and online meetings, and a KDS server.
Figure 4-1  Typical test server setup infrastructure
Table 4-1 shows information that may be useful as a guideline in planning the hardware and software configuration for your KDS environment.

Table 4-1 Minimum requirements for your KDS environment

<table>
<thead>
<tr>
<th>Component</th>
<th>You will need...</th>
</tr>
</thead>
</table>
| Server hardware   | A server machine with at least the following:
|                   | Intel Pentium II TM processor with 500MHz processing speed (800MHz PIII or higher recommended) 512MB RAM (1024MB or higher recommended) Minimum virtual memory at least 2GB, 3+GB recommended
|                   | 20GB minimum, 100GB disk capacity for a full production environment
|                   | The installed components need at least 2GB of disk space.                                                                                                                                                         |
| Server software   | Windows NT 4.0, Service Pack 5, 6a, or above Windows 2000 Server and Windows 2000 Advanced Server 1.0 Domino 5.0.5 for all servers in the existing domain is recommended. The Domain Hub (Domino Directory source) that the KDS server is attached to, should be 5.0.5 and above. |
| Client hardware   | KDS has no client requirement above and beyond what Windows 95 SP2 requires. For K-station users in the Place Manager or Place Designer role who create and/or maintain Places, we recommend at least 128MB RAM; 256MB preferred. |
| Client software   | Operating system Win95, SP2 Win98 SE Windows Me Windows NT 4.0 Workstation, Service Pack 5, SP6a, or above Windows 2000 Professional                                                                                     |
| Browser           | Internet Explorer 5.01 or Internet Explorer 5.5 IE 5.x control in the R5 Notes client.                                                                                                                           |
### Important: KDS is extremely resource-intensive. Memory, CPU, disk and network bandwidth are all stressed to a very high level, particularly during the initial stages when the K-map is built.

Therefore, using the minimum hardware specified in Table 4-1 should only be considered for use in a test environment; otherwise, performance will be dramatically reduced, particularly if a single server is used to perform all tasks.

### Other requirements for your test infrastructure

KDS contains the following embedded products:

- Lotus Domino
- IBM DB/2 Enterprise edition
- Lotus QuickPlace

Lotus Sametime 2.0 (not included in the KDS installer) is required for awareness, chat, and application sharing.

A Domino Master Address Book (MAB), Name and Address Book (NAB), Directory Catalog, or an LDAP Directory server is required for Expertise Location and for the creation of the database People.nsf.

There are many services associated with the KDS server, and quite a few of these services will be familiar to those readers who are familiar with Domino. For readers who are not familiar with Domino, the first seven tasks listed in the following section are regular Domino tasks.

### Services shipped with KDS

1. **Amgr**

   The Agent Manager is a Domino server task that schedules and runs agents on a server-based database. You can run many Agent Manager tasks.
2. Replica

This is the replicator task. It allows the generation of replicas of databases across servers. We will use this task to share information between the Sametime server and the Discovery Server, as well as between Domino and the other servers involved in the test infrastructure.

3. Router

The router must be enabled on all servers that are serving as mail servers, in order to be able to forward (route) the mail messages to other servers which may be Domino or SMTP.

4. Update

The indexer is a server task that enables the updating of views and indexes of database that reside on the Domino server.

5. LDAP

The LDAP server task runs the LDAP server, which performs the lookups for information about users and servers on an Internet directory.

6. DIIOP

- The DIIOP server on the Domino server allows the browser client to use the Domino Object RequestBroker (ORB) server program for data exchange.

7. HTTP

The HTTP Web Server for Domino allows Web browser clients to access data on the Domino server.

8. ATG

The K-map building task builds and maintains the taxonomy.

9. K-map building

This service supports building your organization’s Knowledge-Map by utilizing three main sub-processes: collection, creation, categorization.

10. K-map indexing

This process supports the search feature of the K-map.

11. Profile Source

This server task mines directory sources and supplemental sources to populate the profiles. Some of the information returned from this task is stored in the people.nsf database on the Discovery Server.

12. Profile Synchronization

This task cooperates with the Profile Source spider to manage updates to profiles.
13. NotesSpider
   This KDS server task is responsible for the spidering process of the Notes database, as well as of Notes-related databases such as QuickPlace, Domino.Doc and others.

14. WebSpider
   This server task spiders HTML files on Web servers.

15. FileSysSpider
   This server task hunts file systems and the information stored within them.

16. Metrics Collection
   This task is in charge of collecting and monitoring of metrics data, document weighting, and document popularity, among other tasks.

17. Metrics Processing
   This task handles the processes of collection, calculation, and update that support affinity generation and document valuation.

4.3 The test infrastructure

   A successful installation of the Lotus Knowledge Discovery System (KDS) requires that your KDS servers and client machines meet the hardware and software requirements already mentioned.

4.3.1 What’s needed for your test installation

   For our test we used only two server machines, in order to show you the minimum requirement of machines needed to run a test infrastructure (for example, for test purposes, or for your customers, or even for corporate internal usage, such as presenting KDS to decision makers).

   We set up Domino server R5.05 and Sametime 2.0 software on the same physical machine. We used an IBM PIII 550MHz with 13 GB HDD and 512 MB RAM for this server.

   For the Discovery Server software installation, we used an IBM PIII 800 MHz with 72 GB HDD and 512 MB RAM, and we ran Windows 2000 on both machines.

   For the administration tasks, we used an IBM Thinkpad 600E with Windows NT 4.0 (service pack 5), Internet Explorer 5.04, Lotus Notes R5.05 Designer, and Administration client
System requirements
Figure 4-2 shows the hardware used in our test setup.

Figure 4-2   Hardware configuration - our test setup

4.3.2 Test environment checklist
The following list shows the high-level steps needed to configure Domino, Sametime and KDS for our test environment. We document each step in detail in 4.4, "Installing the first Domino server" on page 82.
1. Set up a Domino infrastructure.
2. Create a people group called KDSAdministrators on the Domino/Sametime server.
3. Install a Sametime server on top of your Domino server.
4. Make changes to the changes to the server documents.
5. Install your KDS software on your KDS Machine; the installation will automatically move to the configuration panel. At this point you do not continue with the configuration itself; instead, execute steps a, b and c now.
   a. Set the ACL of the Stauths.nsf on the Domino/Sametime server, and replicate this to KDS server.
   b. Copy the Server ID of the KDS server generated above to the /data directory of your KDS server.
   c. The Agents that are running on the KDS server need access on the NAB, so you must include KDS/Lotus Notes Companion Products to the ACL of your NAB.
6. Continue with the configuration of your KDS server. After the configuration, you will be asked to shut down the server process.
7. Start your KDS server. (For those familiar with Domino, this step will seem very similar.) On the KDS server prompt, make sure that the HTTP server task started before going to the next step.
8. Start your browser (IE 5.0 or higher), and start the Discovery Control Center.
9. Turn on all services except Metrics Processing.
10. Define your Profile source.
11. Turn on the Profile Synchronizer.
12. Turn on the Metrics Collection.
14. Edit the taxonomy using the K-map editor.
15. Notify users that they will be receiving affinities e-mail.
16. Set the affinity threshold to 50% and enable Metrics Processing.
17. Ask your test users for feedback (after they've monitored their affinities over two or three days).
18. Enable the proposed e-mail repositories.
20. Enable categorization, but without adding additional repositories and without turning on auto extend.
21. Check categorization.
22. Enable Metrics Processing.
23. Decide on new repositories.
24. Disable metrics processing and enable auto-extend (if desired).
25. Add new repositories (only one repository at a time).
26. Verify or relabel the K-map, if necessary.
27. Enable Metrics Processing.
28. Iterate steps 25 through 28 as desired.

4.4 Installing the first Domino server

This section describes how to install the first Domino server in your test environment. (For those already familiar with Domino, you may want to skip to 4.5, “Registering the KDS server and test users” on page 92.)

1. Insert your Domino R5.05 CD in the CD drive of the machine that will be used as the Domino Server, and start the installation process (see Figure 4-3).

   ![Lotus Domino Installation](image)

   Figure 4-3 Starting the server installation

2. Click Next to continue with the installation.
3. We use the Domino Enterprise Server, which allows clustering (that way, if you decide to extend your Domino server infrastructure later on, you’ll only need to add a cluster to your Domino and you’re finished).

Note that each of these options requires a different licensing agreement. Click **Customize** to see Figure 4-5.
4. Figure 4-5 on page 83 shows the extra Domino components you can install. We suggest you choose the Domino as NT Service option. This will make it easy for you to administer your Domino Server remotely, as it allows the Domino Server to start automatically when a machine boots, without physically logging on first. After making your selection, click Next.

5. After the license agreement dialog box, the program files will be copied to the hard disk. When the installation has finished, you’ll see Figure 4-6.

6. Click Finish to end the server installation.

4.4.1 Domino Server configuration

Installing Domino is a two-part process. Part one copies the files to the right directories on the server, and part two is to set up and configure the server.
Figure 4-7  Configuration steps for the first Domino Server

1. As shown in Figure 4-7, select **First Domino Server**.
2. Figure 4-8 shows the two ways of configuring a Domino server. The first is the quick and easy way, while the second allows you to customize the installation to your requirements. Select **Advanced Configuration**, then click the > button on the top right of the Web page, as shown, to continue.
3. In addition to the default-selected, standard Domino services, select the following:
   a. Under Web Browsers:
      - Select HTTP, which will enable Domino as a Web server.
      - Also select IIOP, which will enable this task on the server. (IIOP is needed for all CORBA transactions between the KDS and other Domino servers.)
   b. Click the > button at the top of the screen, as shown, to continue.
4. The information shown in Figure 4-10 is very important, so we’ll split it into mini-sections for easier comprehension.

a. Organizational information

- Enter your Domino domain name in the Domain Name field; this is relevant for mail routing. We used “Demo”.
- Enter the name of your certifier; this is the top-level certifier also known as cert.id. You need this certifier for further registration activities in this domain. Our certifier is also called “Demo”.

Figure 4-10  Configuring certificates and passwords
You can enter a country code if needed. We left this field blank.

Select **Allow Setup to create new certifier ID**.

The Certifier Password field in this section is very important. Enter a password to protect your certifier (this password is case sensitive). We used “password” as this will be a test server with test data. You should use something more secure.

b. Server information

![New Server Identity](image)

- Enter the name of your server in the Server Name field. Our test server is called “Caffreys”.
- Enter the host name of your server; this name must correspond with the network host name of your machine. We used “caffreys.lotus.com”.
- Select **Allow Setup to create a new server ID**.

c. Domino administrator information

![Administrator’s Identity](image)

- Enter the first name of your domain administrator. We used “Dom”.
- Enter the last name in this field. We used “Admin”.
- The ID file that will be generated needs to be secured, so you need a password to secure it against unauthorized usage. We used “password”. Again, you should use something more secure.
- Finally, select **Allow Setup to create new administrator ID**.
d. Customize your network

- Select **Customize...** and click **Edit Ports** to display Figure 4-15.

![Network Options section](image)

*Figure 4-14  Network Options section*

The Port Setup dialog box allows you to configure which type of network ports the Domino Server listens for connections on.

![Port Setup dialog box](image)

*Figure 4-15  Port Setup dialog box*

- We used the TCP/IP port only, so we deselected all other enabled ports by clicking the **Disabled** option to the right of each port.

- Type in the TCP/IP name of the server in the Net Address field. Our server's address is "caffreys.lotus.com", which is registered with the DNS.

- Click the **OK** button at the top right of the screen to complete the configuration of the server, as shown in Figure 4-16 on page 91.
e. Figure 4-16 reflects the information you previously entered, and it allows you to set the default database access for your NAB (see Figure 4-17). It's a good idea to print this page in case you forget a password or server name later.

- Click Exit Configuration to quit the configuration session. It is here that you can define a group or person who will be administering your Domino, and at the same time set “Anonymous” to No Access.

Figure 4-16  Domino configuration successfully completed

- Click OK to confirm your choice.
4.5 Registering the KDS server and test users

So far, we have installed and configured the first Domino Server (which is a plain Domino Server, not a Discovery Server). Next, we need to register the new KDS server and the test users with our new Domino Server. This is done from your Domino Administration client, as follows.

1. Launch the Domino Administrator client, then click **People & Groups**; you’ll see a display similar to Figure 4-18.

![Figure 4-18 Registering Server](image)

- On the right of the screen, select **Registration** and **Server** to display the Register Servers dialog box, as shown in Figure 4-19 on page 93.
Chapter 4. Setting up a successful test environment

Click the **Registration Server...** button to select the server where you want this new registration information to be stored. As we only have one Domino Server at the moment, this will be caffreys/Demo, as shown in Figure 4-20.

Click **OK** once you have selected the registration server, which will return to the dialog box displayed in Figure 4-19.

Click the **Certifier ID...** button and select the /Demo certifier we created during the installation earlier.

Click the drop-down **Security type** field in Figure 4-19 to select your security type. For our test environment, we used the international encryption. The dialog box should now look like that in Figure 4-22 on page 94.
Click **Continue...** to move to the next step.

Figure 4-22 shows the actual server registration dialog box. In the Domino Server name field, enter the name of your KDS server and click the **Register** button. In our test environment, our KDS server will be called Speckledhen.

Note that we have not entered a password for our KDS server. Depending on your own security requirements, you can either leave this field blank or fill in a password. So what are the differences? If you leave it blank, the server will start up automatically when the NT server
boots, without prompting for a password. In contrast, if you do add a password here, the Domino server will not start until you enter a password at the server console. Servers without a password should be kept in a secure, locked room.

2. Registering users

Next, test users need to be added into the system. Referring back to Figure 4-18 on page 92, this is done by clicking the Person button (just above the Server button). The display shown in Figure 4-23 will appear.

![Figure 4-23 Registering users](image)

- When registering users, remember to give them an Internet password, as this is needed for authentication against the Sametime server, as well as against the KDS server.
Once you’ve registered each person, you should edit each person document in the Public Address book (names.nsf) for the purpose of getting Sametime awareness. Modify all person documents to include “Caffreys/demo” in the Sametime server field, as shown in Figure 4-23.

Figure 4-24   Sametime server name entrance in person document

4.6 Install Sametime on top of your Domino server

Before starting this section of the step-by-step walkthrough, you must stop the Domino server. Type the command: q for quit in the Domino console to shut down all the tasks running.

**Note:** Only use the q or quit command to stop a Domino server. Always use the command: tell kds shutdown to shutdown a Discovery Server.

What we’re going to show you now is not the recommended way to roll out Sametime in your enterprise, but since this is a demonstration installation, it’s an effective way of showing Sametime integration.
After the Domino server has been shut down, do the following steps:

1. Start the installation of Sametime and select a complete installation when prompted, as shown in Figure 4-25.

![Sametime Server Installation](image)

**Figure 4-25  Sametime Server install type**

2. The installation program will search for an existing Domino installation and display the following dialog box.

![Question](image)

**Figure 4-26  “On top of Domino” installation?**

3. This is where you learn whether you’re following up correctly, because if this question box does not appear, it means you did something incorrectly. Click Yes to respond and continue.

4. The installation program will copy the Sametime files it needs to your hard disk, and finally, create a few entries in the registry.

5. Once the installation is complete, you’re asked whether to restart the computer. Select No, I will restart my computer later, as this gives you a chance to verify that things are in good order.
6. When you see the dialog box displayed in Figure 4-27, click **OK** to complete the installation.

7. Figure 4-28 shows a check you can perform to see if the Sametime tasks needed have been correctly registered in the operating system. You can reach this screen by selecting the **Services** icon in the Administrative Tools group of the Control Panel.
8. Open the Domino Server document to see the next checkpoint. On the Basics tab of the Domino Server document, verify that the field *Is this a Sametime Server?* reads Yes.

9. Verify that if your Domino Server document now contains a new tab entry called Sametime. This is where you can make some specific settings concerning the Sametime server. (However, we did not make any changes to the default setting on this demonstration infrastructure.)

   This new Domino tab entry is shown in Figure 4-30 on page 100.
4.6.1 Making changes to the server documents

Before we finally get install the KDS server, we need to make some modifications to the various server documents in order for the KDS server to function correctly. Where do you find the server documents?

Open your Domino Directory database and navigate to the server view. Click Server to see the list of servers in your Domino domain (refer to Figure 4-31 on page 101).
10. Edit the Sametime server document and make the following security change:
   a. Select the Sametime server document (in our test environment this is Caffreys/Demo), and click **Edit Server Action**.
   b. Click the **Security** tab.
   c. Navigate to Agent Restrictions.
   d. Modify the field: Run restricted LotusScript/Java agents; enter KDSAdministrators.
   e. Save and close this document.

11. Edit the KDS server document, as follows. (In our test environment, we called it Speckledhen/Demo.)
   a. Select the **Basics** tab, as shown in Figure 4-32 on page 102.
b. Add the group Discovery Servers to the Administrators field. The KDS server setup routine will create this group in the Domino Directory by default.

c. Click the **Security** tab; see Figure 4-33 on page 103.
d. In the Server Agent Restrictions area, add the KDSAdministrators group in the following fields:
   - Run restricted LotusScript/Java agents
   - Run unrestricted LotusScript/Java agents

e. Navigate to the Java/COM Restrictions area and add the KDSAdministrators group in the following fields:
   - Run restricted Java/JavaScript/COM
   - Run unrestricted Java/JavaScript/COM

f. Click the **Internet** tab, as shown in Figure 4-34 on page 104.
g. Click the **HTTP** sub-tab and navigate to the Host Name field under Basics. Here, enter the name of your KDS server (for our test environment, we used Speckledhen.lotus.com, as shown in Figure 4-34).

h. Change the selection box: Allow HTTP Clients to browse database field to **Yes**. (If your KDS server is outside the firewall, set this field to No for security reasons; refer to Chapter 5, “Security” on page 149 for more information).

i. In the Mapping section, go to the Home URL field and enter: `servlet/lp`.

j. Click the **Domino Web Engine** sub-tab and go to the Java Servlets section.

k. From the Keyword fields of the Java servlet support field, select **Domino Servlet Manager**, as shown in Figure 4-35 on page 105.
1. Go to the Character Set Mapping section and change the Use UTF-8 for output field to Yes.

### 4.6.2 Setting the ACL and replicating Stauths.nsf

The Sametime authorization (Secrets) database (stauths.nsf) is responsible for checking your ability to use the Sametime Server with KDS for the purpose of awareness generation. To ensure that this database works, you need to make changes to the Access Control Level (ACL) of this database.

1. **ACL settings for the authorization database.**
   a. Open the stauths.nsf database from the Domino/Sametime server and select **File->Database->Access Control List** from the menu.
b. Click **Add...** and add the group KDSAdministrators as a person group with manager access assigned to them.

c. Set the default access to Reader and enable Write public documents.

d. Close this dialog box and close the Secrets and Tokens database.

e. Open the Domino Directory and navigate to the Servers - Connections view.

f. Create a new connection document to replicate Stauths.nsf from the Sametime server to the Primary KDS Server by using the pull-push replication option. In order for this to work properly, you need an initial replica of Stauths.nsf on the KDS server; this can be done from the Domino Administration Client.

g. Navigate to the Groups view in the Domino Directory and create a people group with the name KDSAdministrators. Populate this group with people who will be maintaining your KDS server, as shown in Figure 4-37 on page 107.

h. Fill the group document with the authoritative administrators in your company; in our case, we call this group KDSAdministrators.
2. Agent Signer: Enter a mixed group called KDS/Lotus Notes Companion Products in the ACL of the NAB with Manager access, all the Roles.

3. Copy the server ID of the KDS server from Domino to your KDS server. When setting up the Discovery Server, the setup routine prompts for the server ID; for a test environment we advise that you copy the ID file from the Domino registration server, where the server and user ID files are created, to the KDS server.

4.7 Installing the KDS software

The last installation step is to install the actual KDS server (did we hear you say “finally”?). If you’ve jumped to this section and have not read the previous configuration section, then go back and read it now. If you think KDS can be installed like any other Domino server, you’re in for a surprise. Paying careful attention to the previous steps in this chapter will mean your KDS server will run smoothly once this final installation is complete.

1. You must log into NT using the name and password of a registered user that has administrator rights to the server. For this test installation, we registered a
user called “Yoda” with the password “yoda” with the proper administrator access rights.

**Important:** The NT user name must be less than eight characters long.

2. You must ensure that Microsoft Internet Information Server is not running already. If it is, the installation process will stop and ask you to remove it, so it’s best to remove it now. To do this, open the control panel, select **Add/Remove programs**, select **Add/Remove Windows Components** and uninstall IIS.

3. Run the file setup.exe on the KDS installation CD. The Installation dialog box is displayed as shown in Figure 4-38.

![Lotus Knowledge Discovery System Installation](image)

*Figure 4-38  The KDS setup screen*

4. Click **Next** to continue and display the Software License Agreement dialog box. Read and accept the IBM License Agreement.

5. Verify your name and your company name, and click **Next** to continue.
6. Enter the drive as well as the directory to copy the KDS files to.

**Tip:** For the best performance, and if it’s possible, place the Program directory and the Data directory on two separate physical disks (for example, C:\Lotus\KDS\Program and D:\Lotus\KDS\Data).

7. Click **Next** to continue.
8. Enter the NT Administrator password twice, and click **Next** to continue. The installation will continue and copy files to the places you specified.

9. When the files have been copied, you'll see the dialog box shown in Figure 4-41.

![Figure 4-41 KDS installation completed](image)

10. Click **Finish** to display the Lotus Product Registration screen.

11. Fill the Product Registration Form and click **Next** to continue.

![Figure 4-42 Installation of KDS data store](image)
12. A final installation process will start automatically as shown in Figure 4-42 on page 110. This is process creates the following:
   a. The Domino log database file on the KDS server (log.nsf)
   b. Initializes the Discovery data store, which in a simple terms creates the DB2 database tables. This may take a few minutes to complete, depending on the performance of your machine.

13. Once the DB2 tables have been initialized, a Web browser is launched that allows you to configure the KDS server as shown in Figure 4-43 on page 111.

   ![Figure 4-43 Setting the primary server type](image)

14. Select the primary server as your server type and enter KDSAdministrators as your master administrator. In our test environment, this person is Dom Admin. We need his name here only because the QuickPlace configuration will not allow a groupname to do the initial setup.

15. Click **Next** to continue.
16. Complete the next Web page as shown in Figure 4-44 by entering the server name we registered earlier in Figure 4-22 on page 94.

17. Enter the TCP/IP registered host name (such as speckledhen.lotus.com).

18. If you copied the server ID file over from the registration server as we suggested earlier, you need to enter the path and filename to this .id file. If you left the server ID file in the Address Book, you need to enter the name of the registration server instead.

19. Enter the name of the registration server in the Get Directory from server field. In our test environment, this server is Caffreys/Demo.

20. Click the Finish button in the top right corner to complete the installation. The progress can be monitored as shown in Figure 4-45 on page 113.
Finally, you’ll see the screen shown in Figure 4-46.

![Creating Knowledge Discovery System Databases](image)

**Figure 4-45  Creating the KDS databases**

21. Click **Exit** to finish.
22. Dismiss the dialog box by clicking OK. Clicking this button will attempt to close the Web browser and, depending on how you have your browser configured, you're likely to see the warning message from Internet Explorer shown in Figure 4-48. Click Yes to continue.

4.7.1 Start your KDS server for the first time

1. Use either the server icon on your desktop, or go via the OS to start the KDS server. Figure 4-49 on page 115 shows the KDS server console as it starts up.

Tip: Watch the server console closely for error messages that occur during the startup. This is the easiest and best way to find any places you may have missed when configuring the server.
Figure 4-49  The primary KDS server console running

2. Start your Browser (IE 5.0 or later), and start the Discovery Control Center by opening the Discovery Administration database (dicoveryadmin.nsf) on the KDS server. To do this use a URL similar to this:

http://speckledhen.lotus.com/discoveryadmin.nsf

3. You'll get a login prompt, as shown in Figure 4-50, that allows you to enter your username and password. The username and password you enter here must be present in the Administrator. Type in your username and password and click OK to start the Discovery Control Center.

Figure 4-50  Logging in to the KDS server

4. Figure 4-51 on page 116 shows the KDS Control Center Web page. From here, all KDS administration tasks can be controlled.
This is the starting point for a KDS administrator, This page is also referred to as the Discovery Control Center and it contains two main sections:

- **Startup**
  - Configure server, Generate Profiles, Affinities and K-map

- **Maintenance**
  - For maintaining System, People, Data
4.8 Configure the KDS services

Figure 4-52 Configure Server and Services

1. Do the following steps to set up services on the Discovery server.
   a. Click **1. Configure Server(s)** in the navigation pane.
   b. Click your server name (in this case, speckledhen/dem); this is a link which opens a services document for your server, as shown in Figure 4-53 on page 118.
1. Scroll down this page; you’ll get to the display shown in Figure 4-54.

![Figure 4-53 Basics server settings](image1)

![Figure 4-54 Services to enable](image2)
3. On this section of the document, do the following:
   a. Enable all the services except Metrics Processing (this is discussed in a later section).

4.8.1 Configuring the spiders

Your KDS server is now configured and running. The next task is to configure and enable each of the spiders in order to retrieve data into KDS.

Define your profile source

The first type of information we need to spider into KDS is the people information. We do this by configuring a People Source document (in our case, we use NAB).

4. Go to 2. Generate Profiles on your navigation pane.
   a. Click Specify Sources.
   b. Select the Domino Address book (if you want to use any other source, then click the Add Source button). In our case, we use Domino.

   Figure 4-55 Specify People Source

   c. Figure 4-56 on page 120 shows the definition of your people source.
d. In the screen shown in Figure 4-57 on page 120 you can fine-tune the field maps (these are important for those using a non-Domino data source or a modified Domino data source).
e. After entering the information and starting the profile source spider on the Discovery Server, your results should look as shown in Figure 4-58.

![Figure 4-58 A completed people source document](http://example.com/figure458.jpg)

**Spidering a Notes database**

Now you need to specify the data source, but before doing this, you should complete the following task.

*Prepare the Help database for better spidering results*

Some Notes databases deliver very nice spider results, such as the Domino R5 Administration Help (found on all full Domino server installations) and the Domino R5 Designer Help database (found on designer clients).

These databases generate very good categories “out of the box” (which is one reason why we decided to use the Administration Help in our case). Another reason for using these databases is availability: anytime you install the Domino Server (see 4.4, “Installing the first Domino server” on page 82), you’ll also have the Administration Help database installed on your machine.

The downside of using these databases is that they do not contain any author information, so tasks such as affinity generation will not work. For this redbook, we wrote a simple agent that added dummy author names to the documents stored in the admin help database. We made this agent slightly “smarter” by adding the author name to documents that were similar in context, rather than randomly.
1. Navigate to the Create K-map section to define your spider source; see Figure 4-59.

2. Enter a title for the data repository (we used “Domino R5 Administration Help.”)

3. In the Type: field, select **generic Notes database**.

4. Select the server where you placed the modified database (we placed it on caffreys/Demo).

5. Select the database from the list.

6. Preferred viewer: is Browser.

7. Enter the host name of the server (in our case, caffreys.lotus.com).

8. You can add more data repositories and find the list of all your repositories.

9. Click the Navigation point **Begin K-map creation** to reach Figure 4-60 on page 123.
10. Click **Monitor Creation Progress**, shown in Figure 4-60, to see whether your spider has started.

11. The K-map building log, shown in Figure 4-61, gives detailed information on the K-map creation process and on what might have gone wrong during the build process.
Refine K-map

The K-map builder builds knowledge maps that may not correspond to the search expressions you are used to in your company, so you must redefine or rearrange some areas of your K-map, as follows:

1. Under 4. Refine K-map, click **Edit K-map** to be able to make changes.

2. As shown in Figure 4-62, you can now make some changes relating to your Discovery Server.

3. Under K-map Editors, click **K-map settings**; you’ll be brought to the display in Figure 4-63 on page 125.
4. Set the first field to **No Maximum**.
5. Set the Maximum number of sub-category levels in K-map field to: **5**.
6. Set the Maximum number of categories a document can be placed in field to: **2**.
7. Click **Add...** to define access group to the K-map; see Figure 4-64.

**Figure 4-63**  Settings for the K-map creation

**Figure 4-64**  Defining K-map access for users/groups

**Adding an additional data repository**
1. Follow the steps above to add a new data repository. Figure 4-65 on page 126 shows we have added a Web spider data type this time, instead of a generic Notes database.
Figure 4-65  Example of a Web spider

2. Figure 4-66 shows the two data repositories, one for a Notes database and the other for a Web repository.

Figure 4-66  Adding more data repositories for spidering

3. To continue, under 4. Refine K-map, click Categorize Data.
Modifying the K-map categorization settings

Figure 4-67 Generating categories

1. From the Categorize Data page, you can make further K-map settings related to categorizing your spidered repository.

2. Click K-map Settings to continue.

Figure 4-68 K-map settings for categorization

3. You’ll need to define the “rules” to be followed when your Knowledge Map is created and categorized.

Enter the values shown in Figure 4-68.

Chapter 4. Setting up a successful test environment  127
4. As you can see, Figure 4-69 is exactly the same log document seen on Figure 4-61 on page 123.

5. This refining section can be repeated as often as you require for your organization.

6. Now we must set the affinity threshold. Navigate to specify settings under 5. **Generate Affinities**, as shown in Figure 4-70.
Affinities are the links that join a person to a particular topic. By changing the threshold, we can either raise or lower the level at which an affinity to a particular topic is assigned to someone.

By default, the affinity setting is very high, which means that unless you are an expert in a particular field, reading and writing a large number of documents in that area, you will rarely see affinities being created. We recommend that you lower the default value.

7. Click Modify Settings to display Figure on page 129. Change the threshold from 100% to 50% for a better affinity calculation result.

8. Make sure the settings you make under the Affinity Approval Policy section go hand in hand with your company’s policy. Your user must be given the right to approve or reject affinities related to him.

9. Save and close the page after making your settings; you’ll see the display shown in Figure 4-72 on page 130.
10. This is the last figure in the Startup section of the Discovery Control Center. We now move to the second half of the control center called Maintenance. After making the affinity settings, you can enable metrics processing—and as shown in Figure 4-73, you can see that the server is now very busy.

11. In order to see if you have received approval mail or not for any affinities, open your e-mail client. You should begin to see e-mails with the title “New Affinities to Approve or Reject” in your in-basket. Figure 4-74 on page 131 shows the contents of one such e-mail.
4.9 The Discovery Control Center (Maintenance)

**Note:** Some of the pages shown in the figures in this section simply display information to keep you up to date concerning the activities of the Discovery Server, and therefore have no associated commentary.

The first page, as shown in Figure 4-75 on page 132, is Servers & Services.
1. System Status, as shown in Figure 4-76, monitors the log results of the K-map building, as well as the metric processing.
2. You may want to add or remove users from the list of administrators; this is the section where this can be done. Click **Modify Groups** and make your changes.

3. The next Maintenance section is the People area, where all user-related information is displayed for administrative purposes. To maintain the
directories you use to capture people's documents, go to People, and click Sources; see Figure 4-79.

4. In the navigation pane, click Deleted to view the deleted people data source you have marked for deletion (if any); see Figure 4-80.
5. If you need to correlate the names of fields in people and profile sources, in the navigation pane click **Profile Field Maps**, where you can add a new field map.

   In the navigation pane, for information about the Affinity Settings page, refer to Figure 4-70 on page 128.

6. In the navigation pane, click **Profiles** to see all your users, as shown on Figure 4-82 on page 136.
The next area is the Data section, which is divided into six pages.

7. In the navigation pane, click **Repositories** to view which repository data type you have spidered so far; see Figure 4-83.

8. In the navigation pane, click **Proposed** to see which repositories are “waiting”; see Figure 4-84 on page 137.
9. If you’ve deleted a repository which has not been finally removed, you’ll see them when you click **Deleted** in the navigation pane; see Figure 4-85.

10. In the navigation plane, selecting the **Field Maps** page, shown in Figure 4-86 on page 138, allows you to define fields that should be mapped for the meta fields when documents are converted to XML.
11. The field created by our agent in the Admin Help database is called KDSAuthors, so we define this field to match the Author meta information; see Figure 4-87.

Figure 4-86  Data Field Maps

Figure 4-87  Global Settings for the Data Field Maps
12. In the navigation pane, selecting **K-map Settings** allows you to exclude certain data types for a specific spider type; see Figure 4-88.

![Figure 4-88 Spider Settings](image1)

Metrics is our next stop on the navigation pane; this is where we modify the settings related to metrics activities.

13. In the navigation pane, click **Settings** to make changes to the default entries. For this demo, we did not make any changes, as shown in Figure 4-89.

![Figure 4-89 Metrics Settings](image2)
14. Figure 4-90 lets you define your own metrics report, which will be similar to the one shown in Figure 4-91.
4.10 Using the Knowledge Map Editor

The Knowledge Map Editor (K-map) is an accessible program that will be installed when setting up the Discovery Server. By default, the installation routine will place an icon on your server's desktop.

If you are working from a remote client machine, you must install the K-map Editor separately by clicking the link **Install K-map software** as shown in Figure 4-62.

After the installation, you can access the K-map data and edit the labels and document placement as desired after a successful login process.

![Login to the K-map](image)

*Figure 4-92* Login to the K-map

You can only gain access to the server if you have been given the access permission through your administrator (i.e. if you belong to the members of a K-map editor group).

Figure 4-93 on page 142 shows you the K-map editor. Here you can see the hierarchical categorization of documents (the taxonomy) in the left pane, and the documents contained in the highlighted entry in the right pane.
From the Editor, you can perform administrative functions on the K-map. Figure 4-94 shows the ability to create a new, remove, or rename a category.

Another way to enter the K-map is through the portal, via the Find button. This will open the K-map, provided you have already created one.
Searching the K-map

Searching “everything about” is the widest search criteria you can define. After clicking the Go button, the documents that match your definition will appear; see Figure 4-97 on page 144.
Once you have completed an initial search, you can refine it in order to narrow the number of documents displayed. Figure 4-97 shows a typical search result screen.

![Figure 4-97   Result of a search](image)

### 4.11 Using the built-in QuickPlace

During installation, the LKDS installation program installs QuickPlace on your machine. In this section we only provide a brief discussion of QuickPlace. For detailed information on using QuickPlace, refer to the IBM Redbooks *Customizing QuickPlace*, SG24-6000 and *QuickPlace Deployment Guide*, SG24-6535. These publications are available at:

http://www.ibm.com/redbooks

The QuickPlace integrated in the KDS server allows you to do the following:

- Create Places
  - Public, or
  - Restricted for certain group of users only
- Create a QuickPlace Web server, accessible via LKDS Server
Figure 4-98  QuickPlace administration Page

The QuickPlace administration Page, shown in Figure 4-98, consists of five sections:

- **Security**
  - Define who can create a QuickPlace, or can administer the server
- **Existing QuickPlaces**
  - Delivers information and monitors the QuickPlace server
- **User Directory**
  - Specify a NAB here for managing QuickPlace members
- **Mail**
  - Defines the mail routing policy
- **Other Options**
  - Global server settings are made here

Figure 4-99  Creating a QuickPlace
1. To create a QuickPlace, as shown in Figure 4-99, enter the name of your QuickPlace, your administrator and his password.

**Important:** The name of the person allowed to administer the QuickPlace server must be a single name and not a group name, and this person must have also been included in KDSAdministrators; see Figure 4-43 on page 111.

2. Confirm the password, enter his e-mail address, and click **Next**.

![Figure 4-100 Confirming the QuickPlace administrator](image)
Figure 4-101  QuickPlace The First Three Steps

Figure 4-101 shows the first three steps (Adding Content, Customizing, and Managing Members) in defining a QuickPlace.

Figure 4-102  K-station List of Places

After creating your QuickPlace, a new place entrance is made; see Figure 4-102 on page 147.
Finally, Figure 4-103 shows what our sample QuickPlace, “The Redbook Place” looks like.
Security

In this chapter, we discuss the following security topics:

- **Security policy**
  The question is always “How can we launch a security policy that we can implement throughout our LKDS infrastructure?” We point out some issues you must consider.

- **Physical security**
  We describe how to restrict physical access to your server and, for those who have physical access, how to enable further restrictions.

- **Network security**
  We describe how to use your network security features to secure your LKDS server infrastructure.

- **LKDS security model**
  - Notes
  - Web

- **How to handle security risks**

- **Secure Sockets Layer (SSL)**

- **Single Sign On (SSO)**
5.1 The security model

Figure 5-1 shows the layers of security points we discuss chapter.

![Security Model Diagram]

**Attention:** In regard to your security policy, it's helpful to recognize that no system is unbreachable. So the question you need to ask is: how easy would it be for an unauthorized person to gain access to your system.
5.2 The security policy

The security to be implemented and adapted on your Discovery Server infrastructure must correspond with the security policy of your company. Before starting, you and your company must determine what your security policy will be, in order to give a clear direction you can follow when implementing Discovery Server.

5.2.1 Physical security

Physical security is concerned with physical access to the servers that data is stored on. There is no point in having a rigid security policy if anyone can simply walk up to a server and log in as an administrator.

The physical security you set up determines which employees can access the server machines, and what actions or activities they can perform. However, there are concerns other than just controlling who can get to a server; once your server is been exposed to an unauthorized person, your data is not secure anymore.

To minimize any damage, you should use the security model provided by some operating systems. These can be used, for instance, to control exactly who can access what software on the server machine. Our advice is to use a combination of these security measures.

Table 5-1 illustrates the security measures that can be taken to secure your server.

<table>
<thead>
<tr>
<th>Security feature</th>
<th>Useful for...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Secure server room</td>
<td>Restricting physical access to the machine</td>
</tr>
<tr>
<td>Hardware locks</td>
<td>Restricting access to computer components</td>
</tr>
<tr>
<td>Operating system</td>
<td>Restricting access to the server console</td>
</tr>
</tbody>
</table>

Secure server room

There are two things to protect when talking about physical security: hardware, and information.

Ideally, all your servers need to be in locked, restricted access rooms, where only authorized users can access them. (Your server is not secure under your desk!)
Hardware locks
The loss of hardware by theft also means a loss of data. For this reason, companies should understand the importance of this “simple” aspect of security.

Hardware locks can be implemented in many ways, depending on the type of equipment used—from rack-system locks to simple notebook locks. Hardware locks can also be implemented on hard disk drives; they can either be built in or (in some cases) removable. If built in, then they are secured with your machine’s physical lock device; if they are removable hard drives, then they need additional steps to secure them.

Operating system security
Some hardware manufacturers, such as IBM, provide software-aided security for their machines. If someone removes the back of a machine, an administrative alert can be sent warning that machine has been tampered with.

Many operating systems also have their own security features, such as locking your screen after a certain period of inactivity, or setting your machine to lock automatically on startup, or only allowing access to the server with a user identification process (user name and password). We advise that you use of all these securities measures where applicable.

Network security
Another way of restricting access to servers is through network security. Network software, as well as the network protocol, can combine to restrict who can access a certain machine, based on their network address and/or network user profile.

With a well-established network infrastructure, you might already use some of these security measures. If that is presently not the case, you should move toward using the full security content of your implemented network.

Isolation network
There are different levels of network security you can implement, but in this case we’ll only focus on one that’s a good starting point for your Discovery Server infrastructure: the isolation network.
This type of network setting, as shown in Figure 5-2, serves two purposes:

1. All data transmitted between the Discovery Servers is on a separate network, making network transmission secure.

2. A great deal of network traffic can be generated between Discovery Servers; by placing them in their own network, the rest of the network is unaffected by potential network slowdowns.

### 5.3 Domino security

The Discovery Server has a lot in common with Domino—including security aspects. All Domino security elements are fully adaptable to your Discovery Server.
In this section we discuss the key areas which we believe will be useful in the security implementation on your Discovery Server. Figure 5-3 shows the security layers of Domino.

Figure 5-3  Domino Security scheme

You can think of security as consisting of several layers; starting from the top and working down, once a user or server passes through one security layer, the next layer of security is enforced. The following sections briefly describe the security layers that you set up to protect the Domino system.

5.3.1 Server security

This is the first level of security that Domino enforces after a user or server gains access to the server on the network. You can specify which users and servers have access to the server, and restrict activities on the server.

If your servers will have Internet/intranet access, set up SSL and name-and-password authentication to secure network data transmitted over the network and authenticate servers and clients. In addition, set up a firewall server to protect Internet servers from unauthorized access from outside the organization’s network (as shown in Figure 5-3).

Security settings in the server document

Your server document provides you with a number of ready-made security mechanisms which you only need to activate, either by making selections or by entering data in security fields (some of which are shown in Figure 5-4 on page 155).
Figure 5-4   Server security settings section

For clarity, in the following figures we focus on specific sections of the server security settings screen.

Figure 5-5 shows the Security Settings section. Use it to implement your overall security on the Domino server.

<table>
<thead>
<tr>
<th>Security Settings</th>
<th>Yes</th>
<th>No</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compare Notes public keys</td>
<td></td>
<td></td>
</tr>
<tr>
<td>against those stored in Directory</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Allow anonymous Notes connections</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Check passwords on Notes IDcs</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Enabled</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Disabled</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 5-5   General security settings section
Select Yes if you want Domino to always compare the public (on user/server authentication) with the ones stored in NAB. This will make all access requests to the server take longer, but will prevent unauthorized users or servers from accessing the server.

Allow anonymous Notes connection must *always* be set to No, so that only authenticated users or servers can gain access to your Domino server.

Figure 5-6 shows the Server Access section. This section is where you must place the names (groups preferred) of those who should be allowed to do specific tasks on the server. The Not access server entrance is the topmost level that will be checked on authentication against the Domino server.

<table>
<thead>
<tr>
<th>Server Access</th>
<th>Who can</th>
</tr>
</thead>
<tbody>
<tr>
<td>Only allow server access to users listed in this Directory:</td>
<td>No</td>
</tr>
<tr>
<td>Access server:</td>
<td>KDSAdministrators</td>
</tr>
<tr>
<td>Not access server:</td>
<td></td>
</tr>
<tr>
<td>Create new databases:</td>
<td>KDSAdministrators</td>
</tr>
<tr>
<td>Create replica databases:</td>
<td></td>
</tr>
<tr>
<td>Allowed to use monitors:</td>
<td>KDSAdministrators</td>
</tr>
<tr>
<td>Not allowed to use monitors:</td>
<td></td>
</tr>
<tr>
<td>Administer the server from a browser:</td>
<td></td>
</tr>
</tbody>
</table>

*Figure 5-6  Server Access section*

The fields can be left empty to allow default settings. If you set a field value, then only those mentioned in this field will be allowed to perform whatever task the field controls.

The last field in this section, Administer the server from a browser, allows you to restrict access to the Domino Web Administrator to administer the server.
In the Agent Restrictions section, shown in Figure 5-7, you specify which Notes users or which Domino servers are allowed to run which types of agents on the server.

<table>
<thead>
<tr>
<th>Agent Restrictions</th>
<th>Who can:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Run personal agents</td>
<td></td>
</tr>
<tr>
<td>Run restricted LotusScript/Java agents</td>
<td>KDS Administrators</td>
</tr>
<tr>
<td>Run unrestricted LotusScript/Java agents</td>
<td>KDS Administrators</td>
</tr>
</tbody>
</table>

Figure 5-7 Agent Restrictions

In the Web Server Access section, shown in Figure 5-8, you can click the dropdown button in order to select a higher Web security access.

Figure 5-8 Web Server Access

Figure 5-9 shows the dropdown view.

Figure 5-9 Setting higher Web security
As you will see later in this chapter, you need a user name as well as an Internet password in order to access the server via a Web browser. This can be made more secure by selecting the Fewer name variation with higher security option listed in Figure 5-9.

Figure 5-10 shows the Passthrough Use view; from here you specify which Notes user and Domino server can access the server as a passthrough server, as well as define the destinations they are allowed to access.

### Table: Passthrough Use

<table>
<thead>
<tr>
<th>Who can</th>
<th>Access this server:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Who can</th>
<th>Route through:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Who can</th>
<th>Cause calling:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Who can</th>
<th>Destinations allowed:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Figure 5-10  Passthrough server setting**

Figure 5-11 shows the Java/COM Restrictions view, where you define which Web browser user group can use the Domino ORBs to run Java or Javascript programs on the server.

### Table: Java/COM Restrictions

<table>
<thead>
<tr>
<th>Who can</th>
<th>Run restricted Java/JavaScript/COM:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>KDSAdministrator</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Who can</th>
<th>Run unrestricted Java/JavaScript/COM:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>KDSAdministrator</td>
</tr>
</tbody>
</table>

**Figure 5-11  Java/COM settings**

Figure 5-11 shows the Java/COM Restrictions view, where you define which Web browser user group can use the Domino ORBs to run Java or Javascript programs on the server.

5.3.2 Database ACL security

After users and servers gain access to another server, you can use the database access control list (ACL) to restrict access that specific users and servers have to individual applications on the server; refer to Figure 5-12 on page 159.

To provide data privacy, do the following:

- Encrypt the database with an ID, so unauthorized users cannot access a locally stored copy of the database.
- Sign or encrypt mail messages users send and receive.
Sign the database or template to protect workstations from automatic evaluating formulas.

**Important:** The database ACL will not be evaluated when the database is not placed on a Domino server. To protect databases, check the box labeled Enforce a consistent Access Control List across all replicas of this database (see Figure 5-14 on page 160).

This prevents users from accessing a database when the database is not on a server, i.e. on a local hard drive.

![Database ACL](image)

**Figure 5-12  Database ACL**

**Important:** Any servers that need to replicate this database must have at least reader access to the server.
The User roles shown in Figure 5-13 allow you further restrict user access. Even if they have enough ACL rights, User Roles narrows the access right.

User roles are application- and database-specific and are added into the design of the database by a Lotus Notes developer. Depending on what the database application is designed to do, a Lotus Notes developer can create their own User Roles and then programmatically allow or deny access to parts of the application.
If you want your Web browser users to have less access rights to the server than Domino users, this is where to set the restriction.

5.3.3 Forms and views security

Although a user may have access to an application, he may not have access to specific design elements in that application, for example, forms, views, and folders. When designing a Domino application, a developer can use access lists and special fields to restrict access to specific design elements.

![Figure 5-15  Restricting a view usage](image)

Figure 5-15 allows you to define who may use a view (provided you have the Designer Client).

![Figure 5-16  Defining who can read or create documents with a form](image)
There are a series of security elements a developer can use, as shown in Figure 5-16. It’s possible to define the group of users who can access documents created with the form for reading purposes, as well as who can create documents with the form.

A private encryption key can be defined to:

- Encrypt documents created with the form
- Disable printing/forwarding and copying documents to a clipboard
- Make the form available for Public Access users

5.3.4 Document security

The restrictions mentioned in Figure 5-16 on page 161 also apply to documents, in some cases. By using or creating “reader” fields to a document, it’s possible to allow or deny people the ability to read a document.

For example, in a Human Resources (HR) application, you would want to allow all people in the HR department to see all documents, but you would want to allow only a manager to view the people they were responsible for.

5.3.5 Paragraph security

Although not a security feature, it is possible to programmatically hide parts of a document a user is viewing; see Figure 5-17. However, this is not a secure way of controlling data, as the information can be viewed from a documents properties box.

Figure 5-17 Security on parts of a document
5.3.6 Field security

Finally, we get down to individual field level control. Figure 5-18 on page 163 shows an example where we only allow people with Editor access the ability to change a field.

![Figure 5-18 Security setting for a field](image)

If we needed to secure this field so that only certain people can view its contents, we can enable field level encryption on those particular fields.

5.4 Web security

You cannot access the Discovery Server without authenticating yourself with the server. How can you secure your server from a unauthorized Web user?

1. Use a secure connection via SSL. (Refer to 5.5, “Secure Sockets Layer (SSL)” on page 174 for a brief discussion on SSL. For more detailed information, refer to SSL documentation.)

2. Use a different subnet mask (network) than your users do.

3. Make use of the extended secured user login of Domino (see Figure 5-8 on page 157).
5.4.1 How security works

Overview
Discovery Server users access its functions from a browser and will be initially authenticated with the Domino Web server by user/password or by SSL.

This authentication will provide a cookie to be stored on the browser for subsequent Discovery Server transactions. For CORBA connections, the cookie will be used to establish the session. For servlets, the cookie will be used to impersonate the user. Ideally there will be a maximum of one authentication per login, but connections to non-Discovery Server resources may require additional logins.

![Flowchart of User Authentication]

Figure 5-19 User authentication

As shown in Figure 5-19 and Figure 5-20 on page 165, there are a number of functions that work hand in hand to make the implementation of security on the Discovery server realizable.
A Discovery Server user may have her account in a domain that is different from the primary domain, and this may be a non-Domino domain (if it is accessible via LDAP). A Discovery Server may be required to access data and to service users from any of multiple domains of multiple types. The Discovery Server will be expected to assemble knowledge data from external Domino and non-Domino domains, and must properly control access to the data obtained from each domain.

### 5.4.2 Authentication and authorization

Another security aspect is the need to define the group of users that will be allowed to perform certain tasks on your Discovery Server. Table 5-2 on page 166 lists and describes the user types you may have.
When authentication and authorization are required
Authentication and authorization are required as follows:

- For opening a Discovery Server portal
- When a user is using a Knowledge Window servlet to access backend information
- For portal access to K-map information via CORBA
- For opening documents that are listed in a K-map query
- For non-portal K-map viewers (for installations where companies use their own portal)

Administrators
- When creating and managing Discovery Server Portals, places, and place membership
- For reading/modifying KDS server settings
- To run the Taxonomy Editor
- For controlling access to specific taxonomy clusters

Server tasks
- For mutual communication between Discovery Server tasks via CORBA (see Figure 5-21 on page 167)

Table 5-2  User types for security

<table>
<thead>
<tr>
<th>User type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Portal User</td>
<td>A user who is using the feature of a Discovery Server portal.</td>
</tr>
<tr>
<td>K-map Viewer</td>
<td>A user who is accessing K-map information.</td>
</tr>
<tr>
<td>KDS Administrator</td>
<td>A person who administers the Discovery Server content map parameters and access rights.</td>
</tr>
<tr>
<td>Taxonomy Editor</td>
<td>A person who controls the taxonomy and can rearrange it.</td>
</tr>
<tr>
<td>KDS server task</td>
<td>A Domino addin task that implements Discovery Server functionality. These may intercommunicate using CORBA/IIOP.</td>
</tr>
<tr>
<td>Primary NAB</td>
<td>The NAB of the domain where you installed the Discovery Server.</td>
</tr>
<tr>
<td>Directory</td>
<td>A NAB or a directory of a non-Domino domain.</td>
</tr>
</tbody>
</table>
Figure 5-21 Communication between a server task (add-in) and LKDS server

Discovery Server tasks intercommunicate by using CORBA/IIOP, and communication between these tasks must be kept secure. For more information on how to keep the communication secure using SSL, refer to “Secure Sockets Layer (SSL)” on page 174.

Figure 5-22 The recognized user type
User cookies
Figure 5-19 on page 164 and Figure 5-20 on page 165 show how cookies are generated. The user cookie is used to track login status set by the Discovery Server so that it is deleted by the browser when the browser exits.

Note: Your browser must be set to accept cookies

Login assumptions
The Discovery Server is an intranet tool, accessible only to an organization’s internal users. Potential users include everyone in the organization, and access is granted based on the user’s standard Notes identification.

In order to use the portal, you must meet two conditions:
1. You must have a Person record in a NAB.
2. You must have an Internet password.

There are four ways to log in to the Discovery Server:
- Enter a URL in the browser's address bar to the Discovery Server system.
- Enter a more specific URL in the browser's address bar to a particular place within Discovery Server.
- Follow a bookmark/favorite to a place within Discovery Server.
- Click a hyperlink in a document to a place within Discovery Server.

Provided you implement single sign-on, users must always authenticate in order to access any part of the Discovery Server. (Refer to “Single sign-on (SSO)” on page 174 for more information on this subject.)

How administrators can allow users to use Discovery Server
Administrator can roll out Discovery Server by sending a link to either the portal, or to the servlet. If you send users a link that will bring them to the servlet, then:
- A user can click the link, get the browser authentication dialog, and type in his user name and password.
- On the backend, make sure you perform any necessary setup for that user, and then take them to their portal.

If you send users a link that will bring them to the portal, then:
- A user can click the link, get the browser authentication dialog, and type in his user name and password.
- On the backend, make sure you perform any necessary setup for that user, and then take them to the portal.
Portal access
Since portal access is dependent on server access, a portal error message will be posted to the Discovery Server and will be registered in the log file. Discovery Server must provide authentication, as well as access control, as follows:

- To any user willing to read the K-map
- To any user who wishes to edit the taxonomy
- To anyone in charge of administering the Discovery Server
- For interserver communications via CORBA/IIOP

This is made possible through:

- Password authentication over HTTP
- Spider acquisition of access right information from the repository domain
- Secure use of CORBA
- Viewer identification over multiple domains
- Viewer access checking before presenting K-map information
- Interserver authentication and security for CORBA operations

Server task authentication and authorization
Communication between Discovery Server tasks will use standard Notes RPC or CORBA/IIOP, depending on the situation. Communication over CORBA will use user name/password authentication with the HTTP server. Notes RPC transactions use standard Domino authentication. This section describes authentication when communication is by CORBA/IIOP.

Authentication and group expansion will use the primary NAB. Server access is limited to servers listed in a predefined access group of Discovery Servers. All Discovery Server tasks are mutually trusted, and the type operations to be permitted will be determined by the server tasks themselves.

K-map viewer authentication
A viewer of K-map information will need to be authenticated to access K-map data. The user will be authenticated by user/password means with the HTTP server running the server that is running the Content Map. The user either must belong to a Domino domain, or to a domain that is accessible via LDAP.

Once authenticated, it will be possible to identify other domains that the user belongs to, so that you can expose documents from those other domains that the user should have access to. In order to provide other domain membership information, user account information for the other domain must be available to KDS.
This account information should be kept in a database or in a section of a database that the user does not have edit access to. (This would be the primary NAB, but could conceivably be in the person.nsf.) The administrator would be responsible for entering this information, and this requires trust of the administrator.

The lack of true authentication to some external domains is tempered by the fact that if the user were gain access where he shouldn't have it, it would only expose document titles, not content. Viewing the documents themselves requires full authentication with the document's domain.

**K-map viewer access control**
From the K-map, the viewer can only see the titles of documents in the knowledge base. To see the actual content of the map, a viewer must access the document separately by whatever document viewer is required, and may have to separately authenticate with the system that serves up the document.

To see the title of a document in the K-map, the viewer must have the same read access that he would be allowed by the system that contains the documents. To accomplish this, the spider will collect the relevant access lists when it spiders the documents, and the access information will be saved in the Content Map. When the K-map is called upon to display a document's title, the access rights list will be checked against the user's context information to determine if the document information will be shown.
Collection of ACL data by a spider

The ACL data gathered by the spider will be of a format specific to the system from which the data was taken and will be opaque to the spider (for example, Notes ACLs are in LMBCS, and NT file ACLs in UNICODE).

Two levels of access are collected:

1) Access rights to the repository
2) Access rights to the document

Repository access rights take into account server access restrictions. Thus, someone who is in a Note DB access list but who is not allowed access to the server will be considered to not have access to documents from the database (see “Server security” on page 154).

This may seem ambiguous, because a user may be able to access the same Notes database on one server but not on another. However, this is standard Domino security. It requires that the KDS server be set up by its administrator to not spider data from an overly restricted server if there is a better choice.

**Important:** Do not spider data from an overly restricted server.
Repository ACL data will be packaged for transmission to the Content Map to be associated with repository-specific information. Document ACLs will be packaged separately and included with other document meta information. ACL "package" will consist of header information in Unicode followed by an opaque ACL byte stream. The document and repository byte streams will be interpreted together by the method appropriate to the repository at the time of access checking. Some access results will be cached with the user context to reduce the time burden of access checking.

This context will be created when the user logs in and will be attached to an ACL access context object in CMap server memory, which will control access checking of all CMap documents. This object and the user context will be removed when the user's session is destroyed.

Access checking

Before allowing a CMap document title to be displayed, the access rights of the viewer must be checked against the ACLs that are associated with the repository and the document. The ACL information is kept in the byte stream that was obtained by the spider. A class like the one shown in Figure 5-24 will be used to do the access checking. There would be one of these for each user CMap session.

This class will be created with a pointer to the Discovery Server user context (previously loaded into memory). For each document, a call to AccessAllowed will be made with pointers to the repository and document packages. Headers of these packages will contain the repository name and type, so that control can be dispatched to the appropriate access checking routine. The repository and document data will contain special flags denoting default or anonymous access, so that detailed checking can be avoided where possible.
Results of the access checking are cached by repository name so that random access of repositories does not cause redundant ACL checking. Some document access caching will be done as well. The return from AccessAllowed() would determine if a document should be exposed or not.

**Administrator and taxonomy editor access**

When a user authenticates with Discovery Server, the user name will be checked against the list of KDSAdministrators and taxonomy editors. If the user belongs to one (or both) groups, appropriate flags will be set in their user contexts. Then this user will be allowed to perform actions that require these privileges.

A Discovery Server user will be authenticated before gaining access to the Discovery Server system, and his access rights will be checked for the function he attempting to perform. Different types of user access are possible, among them user access to the portal, user (viewer) access to K-map data, and administrator access.

**Tip:** The combination of these securities may resemble the graphic shown in Figure 5-25 to secure some important connections on the Discovery Server.
5.5 Secure Sockets Layer (SSL)

Secure Sockets Layer (SSL) connections provide server authentication and privacy of data transmitted through the network. It is optionally possible to have SSL connections provide client authentication.

The SSL is involved in the processes of encryption and decryption. In a Web session, an SSL connection is established directly between the client and the data to be accessed on the destination server. Note that the client and server could use asymmetric key encryption to communicate securely over the network. However, this process is very performance-intensive because of the exchange of the secured data (very large data is encrypted prior to sending, and decrypted on arrival).

To avoid this, the client and server can instead exchange data after they agree on a common encryption key to encrypt the communication. This key is called the session key.

For detailed information on SSL, refer to the IBM Redbooks TCP/IP Tutorial and Technical Overview, GG24-3376 and Java 2 Network Security, SG24-2109.

5.6 Single sign-on (SSO)

Single sign-on is the ability to sign on to multiple trusted systems by logging on only to the first one, which asks a user for authentication credentials. Any subsequent authentications to additional systems are then handled seamlessly in the background.

5.6.1 Recommendations for single sign-on

Following are recommendations by Lotus that you should consider when implementing single sign-on in your organization:

- Upgrade all Domino servers in your environment to Domino 5.05 or above to reap the full benefits of single sign-on.
- Upgrade all QuickPlace servers in your environment that are outside the KDS installation to version 2.06a or above to provide full support for single sign-on. (QuickPlace servers configured within the KDS installation should be left as is.)
Note: Versions of QuickPlace prior to 2.06a do not support single sign-on and interact in unpredictable ways with SSO-enabled servers. Users may not be able to log on to their QuickPlace if the QuickPlace server is an earlier version and they have previously navigated to a server participating in the single sign-on scheme.

- Ensure that all Domino and WebSphere servers that users are likely to navigate to are enabled to participate in the single sign-on environment.

- Only a single SSO realm within a given DNS domain is used because the implementation of single sign-on relies on a cookie that is shared within a single DNS domain.

- Lotus strongly advises against having multiple realms of single sign-on servers within the same DNS domain, because the SSO cookie will be overwritten as a user navigates between realms. In effect, users will be logged in and out of these realms during their navigation. The results are unpredictable and the user might be logged out of a given realm and challenged multiple times. Reaching outside the DNS domain will trigger either additional basic authentication prompts or, if the other domain is SSO enabled, will overwrite the cookie and cause token validation problems.

- Change the default token expiration value for SSO token to be a time period equivalent to a working day (480 or more minutes) to prevent any user confusion when the SSO cookie expires. Also ensure that users understand that they will need to restart their browsers just before the expected token expiration to forestall the interruption of program access.

- Normally when the SSO cookie expires, the browser will open a new login prompt in the same browser window. However, the effect varies depending on what the user was doing at the time.

In many cases, the user may not be able to fully recover and return to what they were doing. This is likely the case if the cookie expires when a user opens a modal dialog or when saving changes to a place. It's possible that the browser may lose context information when challenging the user or opening the login prompts in new windows.

To recover from such cases, it is best to close down the browser and restart the browser session. To minimize such confusion, it is best to ensure that the cookie expiration length is longer than the length of a typical browser session.

- Ensure that Anonymous access to the LDAP port is enabled on the KDS server.

If such configuration is not done, it is likely that users will receive a second authentication prompt when accessing any of the Notes Mail or Notes View portlets. Additionally, it is likely that users will be unable to browse servers and database listings in the portlet server picker. Typically an error message will be returned, informing the user of this situation.
Disabling single sign-on on the KDS server

If you need to disable the single sign-on environment, follow these steps:

1. Using the Domino Administrator program, open the KDS server document in edit mode.
2. Click the Internet Protocol tab, and change the Home URL to: /servlet/lp.
3. Click the Domino Web Engine tab and change the HTTP sessions - Session authentication option to Disabled.
4. Click Save, then click Close.
5. Open the notes.ini file and delete the following line:
   LPSSO=/lp/lpmain.nsf/fa_login?OpenForm
6. Restart the Discovery Server.

5.6.2 Single sign-on issues

Using X.509 Certificates

To use X.509 Certificates for authentication in Discovery Server, your organization needs to enable session-based authentication on the Discovery Server.

Follow these steps:
1. Install and distribute X.509 certificates to the user base.
2. Create a Web single sign-on configuration document on the KDS server.
3. Enable session-based authentication on the KDS server, either single-server or multi-server.
4. Modify the server’s home page to be the login form:
   /lp/lpmain.nsf/fa_login?OpenForm
5. Enable client certificates as an authentication option for the SSL port.
6. Disable name and password as an authentication option for the SSL port.
7. Disable anonymous as an authentication option for the SSL port.
8. Require users to connect using SSL. That is, log in to KDS using:
   https://server.com/
**Keep SSO enablement homogenous for all servers configured for KDS**

All servers in the KDS must have SSO enabled and disabled uniformly. If SSO is enabled on the Domino hub server, then it should be enabled on the KDS primary and secondary servers. Don't mix SSO and non-SSO servers in your KDS configuration.

**SSO and Secure Sockets Layer (SSL)**

This note applies only to Domino servers (Versions 5.0 to 5.03) that are configured to run only under SSL.

When configuring remote Domino servers (Versions 5.0 to 5.03) to be accessed from KDS servers, follow the instructions for remote Domino servers running only under Secure Sockets Layer (SSL).

If a KDS server is configured for single sign-on (SSO), the following may occur:

- Users may experience a second authentication prompt when viewing Calendar portlets from remote servers running versions of Domino from 5.0 to 5.03 that are configured to run only under SSL.
- Users may receive error messages when attempting to browse databases and views using the server/database picker.

**Expired SSO tokens block access to Discovery Server and K-station features and functions**

If you're experiencing difficulty using KDS enabled with single sign-on, your SSO token might have expired. When your SSO token expires, you'll need to reauthenticate in order to gain access to some of the KDS features (for example, browsing and searching in the K-map, or spidering data repositories).

Certain actions prompt you to authenticate, and others do not. If you are not prompted to reauthenticate, you can be running KDS with incomplete functionality.

When an expired SSO token blocks access to KDS features, you're apt to receive any one of the following messages:

- Cannot retrieve directories for the server. Please type in the directory, database name, and view name.
- Javascript error: Object required
- Error: Can't move focus to the control because it is invisible, not enabled, or of a type that does not accept the focus.
- Error on console:<date/time stamp> Authentication Failed for <person name/server>
Addin: Agent error message:

If you experience functional breakdown caused by an expired SSO token, try the following:

- Close your browser to log out of the product and re-enter in edit mode.
- Avoid leaving an unsaved form idle in edit mode for a lapsed time of 10 to 30 minutes if KDS is running with SSO enabled.
- Configure the SSO token expiration time to handle a typical time interval for a KDS session, and communicate this time constraint to KDS users.
- We strongly recommend that the token expiration value for the SSO token be set to a time period equivalent to a working day (480 or more minutes) to prevent user confusion regarding when the SSO cookie expires.
- Refer to the install & setup help under the topic “How single sign-on works in a K-station Only implementation” for instructions on how to change the SSO token expiration time.
- See also the detailed guidance provided in 5.6.1, “Recommendations for single sign-on” on page 174.

With SSO enabled, the Add Portlet dialog is blank after you authenticate into another SSO domain

This problem occurs because the second SSO domain overwrites the K-station domain's SSO LPTA token.

Ignore spurious K-station error messages at server console during startup

During KDS server startup, with SSO enabled, the server console displays the following messages:

- Server document: The Home URL has not been set to the proper value. To run K-station, you will need to type the full URL:
  
  servername.domain.com/servlet/lp

- Errors were found in the server document. Please correct these errors before running the K-station.

- In the KDS server document in an SSO environment, the Home URL is lp/lpmain.nsf/fa_login?OpenForm and not /servlet/lp. You can ignore these messages because K-station will load properly without adding /servlet/lp to the server name in the URL.
SSO-enabled users should open K-map only via the K-station Find command

SSO-enabled users cannot access K-map by entering a URL (for example, http://yourserver.acme.com/kmap.htm) directly in the browser address bar. (Users that try to do this will see a JavaScript error message.)

To open K-map, users must click Find within K-station.

Additionally, if users with Internet Explorer 5.5 try to open K-map via URL, they will receive the error message: Problems with Web page...

If SSO is enabled, the default affinity-notification message should be changed

To change the default e-mail message that KDS sends users when affinities have been proposed for them to approve or decline, do the following:

1. Remove the profile link at the bottom of the message.

2. Add text to the message that tells users they should first log into K-station and then open their profiles, so they can approve or decline proposed affinities, view K-map categories that correspond to proposed affinities, and so on.
Performance, server sizing and scaling

This chapter will help you to plan your Discovery Server environment, ensure sufficient performance for all Discovery Server users, and tune and monitor the server's performance.
6.1 Planning a Discovery Server system

The Discovery Server is a very complex system, and planning a deployment to a large number of users entails a significant amount of forethought. Aside from the planning requirements, if you are going to use the full Knowledge discovery system with K-station as the portal, there are even more requirements to be evaluated.

**How many users are planned to access the DS concurrently**

It is very important to get an idea of how many users will be using the Discovery Server system concurrently. Generally, about one-third of all registered users in a company will actually use a certain network service (e.g. Discovery Server) concurrently (although this may differ in your environment). This generalization may vary during “rush” hours (for example, at 9:00 AM, every client on the net is checking and replicating mail, and so on).

The next important thing is to identify the approximate maximum number of documents your taxonomy will be built of (up to 10 to -15% inaccuracy is allowed). This is necessary because the size of the taxonomy is not only a matter of required disk space, but also a matter of processing power and network backbone capacity.

6.2 Distributed Discovery Server services

If you plan to have more than 15 persons using a K-map of more than about 10,000 documents, you will need to distribute the Discovery Server services to different servers, enabling them to perform better when running multiple spiders, multiple workers etc. to concurrently serve user queries.

The Lotus Discovery Server has two types of server: a primary and (multiple) secondary Discovery Servers.

The primary server does the following:

- Controls all secondary servers
- Schedules tasks, workers and spiders
- Starts and stops all services and tasks on the secondary servers
- Runs all DB2 tasks (DB2 stores K-map)
- Maintains and monitors all queues (except the private ones which the spiders need when spidering a certain data repository)
- Allows you access to schedule and set all tasks, spiders, settings and repositories
- Allows you access to monitor activities and log files
- Holds all databases to be replicated to the secondary servers
- Maintains the “dashboard.nsf”

The secondary servers should be used for the following tasks:
- To run the K-map indexing and K-map building tasks
- To run spiders
- To run K-map building, Metrics Collection and Metrics Updating
- To perform the K-map queries the users issue

A distributed implementation of Discovery Server allows two or more servers to share the workload of the deployment by running only some of the required services and tasks. Each distributed environment is composed of the following:
- A primary server - where master replicas of all databases live. Only one primary server can exist.
- Secondary servers - running any or all of the following services:
  - K-station
  - Profile Synchronization
  - K-map Indexing
  - Metrics Collection/Updating
  - Spiders

You need to know how a given Discovery Server service impacts your resources in terms of performance. Table 6-1 provides a rule of thumb regarding which LDS services make use of what resource to what extent:

<table>
<thead>
<tr>
<th>Service</th>
<th>CPU</th>
<th>Network I/O</th>
<th>Memory RAM</th>
<th>Memory Virtual</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spiders</td>
<td>low</td>
<td>intensive</td>
<td>medium</td>
<td>medium</td>
</tr>
<tr>
<td>K-map Indexing</td>
<td>intensive</td>
<td>intensive</td>
<td>intensive</td>
<td>medium</td>
</tr>
<tr>
<td>K-map Creation</td>
<td>intensive</td>
<td>medium</td>
<td>medium</td>
<td>medium</td>
</tr>
<tr>
<td>Metrics</td>
<td>intensive</td>
<td>medium</td>
<td>intensive</td>
<td>intensive</td>
</tr>
</tbody>
</table>
Using these assumptions, early load tests, and recommendations based on the knowledge of data flows between the Discovery Server components, Figure 6-1 shows the possibility of having a dedicated server for every Discovery Server component, and how to distribute the services between servers.

Note: The example assumes you use K-station as the portal interface for Discovery Server.
As you can see from Figure 6-1 on page 184, this would use a large number of expensive servers. If you do not have the possibility or do not see the need to have a dedicated server running a single Discovery Server task, we recommend the following deployment considerations:

**Primary server:**
- Scheduler
- Metrics Collection

**Secondary server 1:**
- Metrics Processing (Calculation, Updating and Reporting)
- Profile Synchronization
- Profile Source Spider

**Secondary server 2:**
- K-map Building
- K-map Indexing
- File System Spider
- Web Spider
- Notes Spider

If you only want to run two dedicated machines, providing all Discovery Server services, we recommend aggregating the services of Secondary server 1 and Secondary server 2.

For any number above three dedicated servers, we recommend distributing service to a dedicated server in the following order:

1. Enable any spider on a dedicated server.
2. Continue enabling any other task on a dedicated server, following the individual needs of your infrastructure:
   a. If you have a large directory repository with numerous updates every day, distribute profile-related services to a dedicated machine. Profile Synchronization, Profile Source Spider and Metrics Processing should run on the same machine.
   b. If you have huge numbers of data repositories, with a great number of updated documents every day, distribute K-map-related tasks (K-map Building, K-map Indexing) to a dedicated machine and spiders on another dedicated machine. Try to have Metrics Indexing on a separate machine, since it is very CPU- and RAM-intensive.
Typical spider rates on our test machines (4-way 200Mhz Pentium CPU, 2 GByte Ram, 100 MBit NIC) with notes databases on separate Domino Servers (1 CPU 700 Mhz, 512 MByte, 10 Mbit NIC) are about 4000 docs/hour. This value is for one single spider thread; it may differ in your environment, depending on server speed and network capacity.

If your repositories are taking too long to be spidered, consider adding another spider server if the problems you are experiencing are a result of the machine being unable to cope on its own with the repositories to be spidered.

If the problems are a result of the connection with the repository being too slow, try to create a replica of the Notes database (if that's the type of repository you are experiencing problems with) or a mirrored copy of a Web site on a local machine within your intranet. Spiders usually need high I/O performance capabilities, so any part of the server that can be enhanced in terms of I/O performance is involved (for example, gigabit Ethernet adapters and high performance disk arrays).

Note: Spiders only spider new or changed documents. They will only spider all documents if told to do so. Once your K-map is created, you will not need as much performance as you did when initially building your K-map and spidering the repositories, because the spiders will only spider the delta between old documents and all documents.

To spider and test new repositories, you should have a small test environment to see if the documents fit into your taxonomy well.

c. If your users use the K-map frequently, and change their topics of interests frequently, be sure to enable Metrics-related tasks (Metrics Collection, Metrics Processing) on dedicated servers. Metrics services hosting servers usually need high processing performance.

Note: With Discovery Server 1.0, we recommend always enabling the Metrics Collection Service on the primary server.

Based on the requirement to have all users access the K-station portal interface through one single point of entry (which makes it impossible to have different dedicated K-station home servers for all users), we suggest you use a so-called “IP-Sprayer” (for example, Internet Cluster Manager or ICM), which balances the load between identical K-station servers.
To ensure data integrity and consistency between the K-station servers, we recommend putting them into a Domino cluster to enable instant replication. (Note that this suggestion has not yet been tested in a large, real-life environment. If you do not need a single point of entry to your portal environment, you might as well use dedicated K-Station home servers for a certain number of users, or users that are a community in a certain geographical location.)

If you have seven machines available and want to use two machines as K-station servers (or other portal servers), try to have the following configuration:

1. **Primary server:**
   a. Metrics Collection

2. **Secondary server:**
   a. Profile synchronization
   b. Profile source spider
   c. Metrics processing

3. **Secondary server:**
   a. K-map building

4. **Secondary server:**
   a. K-map indexing

5. **Secondary server:**
   a. Spiders (Notes, File, Web)

6. **Secondary server:**
   a. K-Station - do not enable any Discovery Server tasks

7. **Secondary server:**
   a. K-Station - do not enable any Discovery Server tasks

### 6.3 Server hardware considerations

In addition to the distribution of services, there are several considerations regarding the server hardware to be used:
6.3.1 Network interface cards

Since all processing of data repositories and people repositories is done communicating with the primary server—which is the only server in your distributed environment (in the version 1.0) serving the K-map (DB2)—it is strongly recommended you have high performance network adapters (preferably Gigabit Ethernet) installed in the servers. It is best to have two NICs installed in a server, one for LAN connection and one for backup system connection (for more information, refer to 6.3.5, “Network considerations” on page 189).

6.3.2 Memory size requirements

The size of the RAM in the Discovery Server is very important. Most Discovery Server services are highly memory-intensive, especially Metrics Processing, K-map building, and the spiders. We recommend RAM sizes of about 512 MB up to 1024 MByte per CPU. This is especially true for the primary server and the server running the K-map Indexing task—each should have 4 GB of RAM, since the memory load on these machines is very high.

6.3.3 CPU performance

For a productive Discovery Server environment, we do not recommend running any of the Discovery Server services on CPUs below 650 Mhz Pentium III Xeon level (or other CPUs with lower performance than that).

Furthermore, you should understand that no matter how many CPUs your server has, every service instance processes only one repository at a time. It is not possible to have two service instances process one repository concurrently.

Therefore, we offer the following rule of thumb for the number of services per CPU: for every CPU your servers have, you can have two instances of a particular service enabled (which is also the default value for every service).

Note: If you are running the Notes spider and the Web spider on a 4-way server, you can have, for example, 8 concurrent Notes spiders and 8 Web spiders enabled on that machine to spider data repositories.

Although the Discovery Server services automatically use all processors in a SMP server, you should always run performance logs to see how well the CPUs are used.
6.3.4 Storage requirements

You need to properly size the disk storage of your Discovery Server system (and this task depends on what your data repositories are like, how many data repositories you think you will be using, and so on).

It’s very important to have enough disk space available on the primary server, because the primary server stores all queue files and DB2 databases. The queue files can grow large, especially if no worker task is working on them.

If you spider a large Web site and keep the consumer tasks (for example, Metrics collection, K-map indexing, and so forth) disabled, the queues will grow until there is no space left on the drives (in version 1.0). The permanent data storage for the K-map is the DB2 database.

It is not possible to estimate the exact size of the K-map and Metrics in DB2, since this data depends on the size of the documents you spider. However, the space needed for one document in your Metrics and K-map database is probably about an average of 30 KBytes, while some tests have shown that the required disk space needs to be 1.5 times larger than the original repository to spider. This also includes the people resource databases.

The disk size requirements for secondary servers are not as high as they are for the primary server.

To be able to dynamically extend available disk space, consider using consolidated disk storages for all servers (e.g. IBM Shark arrays), where you install the executables and data stores of every Discovery Server, while installing the operating system to a local hard disk.

6.3.5 Network considerations

The servers in a Discovery Server environment communicate frequently with the primary server, since it does all the scheduling and data storing. The communication protocol used is IIOP.

Due to frequent communication between the Discovery Servers, you should try to put them in a dedicated, isolated subnet in order to maximize the speed between the servers and minimize the network traffic on your users network. We recommend you have a switched connection between the servers.

If your data repositories change on a frequent basis, with a considerable number of changed and newly created documents, it is inadvisable to separate your Discovery Servers and connect them with slow links (2 MBit backbones or less). It is also inadvisable to distribute a Discovery Server environment to distant geographical locations if the WAN connection to these locations is slow.
Spiders perform better when they are connected \textit{directly} to the primary server via high-speed connections (rather than having a high-speed connection to the repository to the spider, and a slow link to the primary server).

If possible, always have the fastest link to the other Discovery Servers. If you can, place a Domino Server serving the Notes databases to be spidered in the same physical network or LAN (as this is more desirable than to spider a repository via a backbone that is already heavily used by other servers or users).

If you can afford to have reduced security on interserver communication, avoid tunneling and encryption between the servers; encrypted data contains an overhead (the encryption) over non-encrypted data.

In terms of K-station performance, it’s best to have fast connections between the K-Station servers and the Domino servers that provide the Notes View Portlets with view content.

Extremely high-load network operations like backups should be done nightly via mirroring to disk array, and storing the contents of the mirrored servers to an archive (tape, optical) asynchronously. In Discovery Server v1.0, the servers have to be shut down before you perform backup operations. Shutting down the servers forces the services to save their current state to the queues. Without shutting down, the current state of the spiders would be lost.

This procedure enables you to have minimum downtime, because the mirrored images of the servers can be created in virtually no time, while more time-consuming operations like storing the mirrored servers to tapes or disks can be done when the main Discovery Server is already up and running again.

In release 1.0, you need to shut down the servers before backing them up by using the Discovery Server console command: \texttt{tell KDS shutdown}. You must shut down all secondary servers first, before shutting down the primary server.

6.4 OS performance tuning considerations

In terms of tuning operating system performance, the following sections contain general Windows NT and Windows 2000 tuning tips. (All settings should only be done by trained system administrators and IT professionals.) Keep in mind that destroying your server’s registry may lead to a major problem, requiring a reinstallation of your operating system.
6.4.1 Tuning Windows

Based on the assumption that your Windows server has more built-in memory than 128 MByte (which is strongly recommended), the following Windows registry settings should be taken into consideration:

1. "HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Session Manager\MemoryManagement":
   a. "DisablePagingExecutive" = 1: no parts of the Windows kernel are swapped to the page file
   b. "IOPageLockLimit": set to a decimal value of “RAM size in kBytes divided by 8”

2. "HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\PriorityControl\Win32 PrioritySeparation" should be set to “0”.

<table>
<thead>
<tr>
<th>Registry Key</th>
<th>Default Setting</th>
<th>Recommended Settings(decimal)</th>
</tr>
</thead>
<tbody>
<tr>
<td>IoPageLockLimit</td>
<td>0</td>
<td>1024-2048(16MB) 2048-4096(32MB) 4096-8192(64MB) 8192-16384(128MB) etc.</td>
</tr>
</tbody>
</table>

Table 6-2: Optimal IOPageLockLimit size

   c. LargeSystemCache = 0: reserves a smaller part of the page file for the system
6.4.2 Optimizing virtual memory

The pagefile is used to free physical memory by writing (paging) memory to disk. It is used as virtual memory when your application experiences a lack of physical RAM. However, this is undesirable for overall application performance, since the virtual memory resides on the machine’s hard disk, which is much slower than physical RAM. The question of what the optimal size of your pagefile should be is controversial, but there are a few rules of thumb, as follows.

Set the minimum and the maximum size of the cache file to an equal value, to avoid fragments in the file when new space has to be allocated. When your server has more than 512 MBytes of RAM, the size of your pagefile should be about 2 to 3 times the size of your server’s physical RAM. (This applies when running Discovery Server. For running other services, this value could be different.)
The identification of the location of the pagefile is a complicated decision, and is specific to the number of hard drives. There are a number of ways to configure this, with each solution giving you a different level of performance. Table 6-3 lists some suggestions.

**Table 6-3  Optimal pagefile location**

<table>
<thead>
<tr>
<th>Location</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Optimal</td>
<td>Put a paging file on a hard disk (or on multiple hard disks) that does not contain the NT system files or on a dedicated non-RAID partition.</td>
</tr>
<tr>
<td>Good</td>
<td>Put one paging file on a stripe set with no parity.</td>
</tr>
<tr>
<td>Adequate</td>
<td>Put one paging file on a stripe set with parity.</td>
</tr>
<tr>
<td>Worst</td>
<td>Put the paging file on the same drive as the NT system files.</td>
</tr>
</tbody>
</table>
Do the following:

1. Right-click My Computer and select properties. Click the Performance tab.
2. Click Change. Click the appropriate drive.
3. Enter the initial and maximum size and click Set.
4. Repeat for all appropriate drives.
5. Reboot the PC.

File swapping performance is improved in relation to the number of physical hard drives used. Note that you must have multiple physical hard drives to take advantage of this feature. Simply spanning pagefile.sys across multiple logical drives will have the opposite effect and will reduce performance on the system.

**Note:** Always try to have Discovery Server program files and data files installed on a physical hard disk drive other than the drive where you installed the operating system.

### 6.4.3 Tuning Windows

The following sections will help you increase the general performance of your Windows operating system.

#### Network

As you probably know, every networking protocol installed on your machine causes the associated network search service to run, producing a considerable amount of traffic on your network. To optimize your operating system's network performance, avoid installing unused network protocols.

List the protocols by the frequency that they are used (for example, the most used protocol should be first and the least used protocol should be last):

1. Right-click Network Neighborhood.
2. Click Properties.
3. Click the Bindings tab.
5. Re-order the appropriate protocols.
7. Re-order the appropriate protocols.
8. Click OK.

Data transfer via TCP/IP can be significantly increased in networks with high bandwidths and long latency times by increasing the size of the packets. To achieve this, add the registry key “TcpWindowSize” (DWORD) to:

“HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\Tcpip\Parameters”
This value can be changed up to a maximum of 65535 Bytes. The default value depends on the NIC being used and is usually about 8000 Bytes.

The size of the TCP window controls the bytes the system can retrieve before it sends out an acknowledgement. If you change this value, you should change it to an even multiple of the Maximum Packet Size (MSS). In an Ethernet network, the TcpWindowSize value defaults to 8760 Bytes, which is six times the value of a 1460 Byte segment.

**CPU**

The Windows NT usage of the CPU's level 2 defaults to 256 KBytes. If your server's CPU has more built-in level 2 cache, you should increase this value to significantly improve your system's performance:

Registry Key:
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Session Manager\Memory Management]

Value Name: SecondLevelDataCache

Data Type: REG_DWORD

Data: Cache (in Kb) using Decimal Notation

In Windows NT 4, you can watch the CPU load using the Task Manager. In Windows 2000, use the Performance monitor in the Start/Settings/Control Panel, as shown in Figure 6-4.

![Starting Windows 2000 Performance Monitor](image.png)

Figure 6-4  Starting Windows 2000 Performance Monitor

However, this does not actually monitor whether the CPU is really overloaded—for example, a CPU showing 100% load is not necessarily overloaded. Much more important is how long a thread has to wait before it gets to use the CPU. To monitor this value, you should use the system monitor.
6.4.4 Monitoring performance

To improve your Windows system's performance, consider monitoring some system resources on a regular basis.

To monitor the CPU load and whether the CPU is overloaded or not, and to monitor if your server has enough physical memory and hard disk space available, simply start the Performance monitor (in Windows 2000) and right-click in the graph window area of the Performance monitor window; see Figure 6-5.

**Figure 6-5 Performance monitor startup screen**

To monitor the CPU load and whether the CPU is overloaded or not, and to monitor if your server has enough physical memory and hard disk space available, simply start the Performance monitor (in Windows 2000) and right-click in the graph window area of the Performance monitor window; see Figure 6-5.

**Monitoring CPU load**

To check if the CPUs in your server are too slow or if you need more CPRs in an SMP system, you can monitor the load on your CPU by right-clicking in the graph area and selecting Add.
In the dialog box, select the object **System** and the source **Processor Queue Length** (refer to Figure 6-7).

If the average value is constantly above 4, this is a good indicator that your CPU is overloaded. The length of this queue is the most important factor in identifying a CPU that's too slow.

**Note:** The queue counter only runs correctly when you enable another indicator of the thread-object in the system monitor.
To see how much time of the overall CPU performance is used for un-idle tasks, the best indicator is the object Processor with the indicator %ProcessorTime (refer to Figure 6-8).

If the biggest part of this graph is above 80 percent, a faster CPU is the required solution.

**Monitoring disk performance**

To be able to monitor hard disk performance under Windows NT4 with the system monitor, this service has to be started first. Open a Windows command line and enter the command `diskperf -y`. A reboot of the system will be required.

**Note:** Only enable the disk performance monitor for the time of the actual performance test, since the diskmonitor consumes a considerable amount of processor time. Disable the service with the command `diskperf -n`.

If your system constantly shows high disk activity, this is likely to be due to insufficient physical RAM. Consider a RAM upgrade of your system. A good way to monitor whether your server’s physical RAM is too small is to watch the number of pages with the system monitor. The data source is the object Memory, where you should monitor Page Faults/sec and available bytes.
Add counter “Page Faults/sec”

Figure 6-9 and Figure 6-10 on page 199 show the addition of two memory performance counters.

Add counter “Available Bytes”

Figure 6-10
The first indicator tells you how many pages a second the system has to perform in order to supply all threads with the required data. An average value of about above 100 is an indicator of insufficient physical RAM. If 100 or higher values occur as peaks (few seconds to minutes), this is not an indicator of insufficient RAM.

6.4.5 Performance considerations in general

Figure 6-11 on page 200 shows a typical Discovery Server demo installation. Note that all the average values are pretty bad, and the system is over the peak of its processing capabilities when put under stress.

![Performance](image)

Figure 6-11  The load on a demo installation of Lotus Discovery Server

In order to be able to find performance bottlenecks, consider establishing performance log files that are saved to disk and which cover at least a few days of system performance.

If you are experiencing problems with a certain Discovery Server service, add this service to your log and monitor the load that this service produces in the context of the counters we previously discussed.
Figure 6-12  Webspider on already overloaded single CPU LDS demo system

If you notice a particular task consuming too much memory or requiring too much CPU power, you are in a better position to decide if you have to distribute this service to another server or if you have to upgrade your server.

Figure 6-12 shows a very drastic example of what a performance graph looks like when you add important processes of your Discovery Server system to the performance monitor.

The screenshot was made on a single PIII-800 CPU, 384 MB RAM, all-in-one Discovery server (or rather a misused IBM PC300 PL, a workstation not designed for server purposes) running all tasks at once. It had to cope with about 20,000 data repository documents, as well as 50,000 people repository documents and had just been enabled to spider a Web site four levels deep. The webspider task is the bold line in the lower white section of the graph, measuring % Processor Time. Don’t underestimate how resource-intensive Discovery Server can be.

**Tip:** To be able to better distinguish between tasks on a crowded performance monitor screen, click the light bulb above the graph area and switch between counters, highlighting them in white.
6.4.6 General tuning recommendations

To tune some Windows system settings on your particular server, you actually have to monitor some system resources with the performance monitor. Others apply to all systems running Discovery Server.

**Primary server (Windows NT 4)**
For the best performance on the primary server, we recommend that you set the server service to “Maximize Throughput for Network Applications”.

**Secondary server A (Windows NT 4) spidering a file system**
For the best performance on a secondary server, we recommend that you set the server service to “Maximize Throughput for Network Applications”.

Also, set the spider process priority to High.

**Secondary server B (Windows NT 4) Webspidering**
For the best performance on a secondary server, we recommend that you set the server service to “Maximize Throughput for Network Applications”.

Also, set the spider process priority to High.

6.4.7 Performance-related Notes.ini settings

KDS_BOA_THREADS should be no less than 20. A value of 60 seems to utilize 1-2 processors well. Essentially there is no upper limit, but you should monitor whether a multi-processor machine is utilizing all its processors as much as possible and then increase this number to boost the utilization. There is probably a point where increasing the number won’t increase actual processing efficiency because other factors like IO bandwidth will begin to dominate.

KDS_EVENT_THREADS should be no less than 6 when KDS_BOA_THREADS is 20, and should be 18 when KDS_BOA_THREADS is 60. KDS_EVENT_THREADS should always maintain this proportional relationship to KDS_BOA_THREADS.

6.4.8 Geographical server distribution

If you cannot afford to replicate certain databases to a server near your Lotus Discovery Server, where the connection between the LDS servers and the regular Domino application server is best—or if you do not wish to do so for some reason—you need to be aware of some problems, especially when you decide to have remote LDS servers in a different geographical location than the primary LDS server.
Spiders write all the data they retrieve from their repositories directly to the LDSFullText-, LDSMetrics- and LDSTaxonomy work queue. These queues reside only on the primary server (in LDS V1.0). Be aware that there is a considerable amount of network traffic generated between the primary and the secondary server. In fact, the amount of traffic between the secondary Lotus Discovery Server and the primary Lotus Discovery Server is at least twice the amount of traffic generated between the secondary Lotus Discovery Server and the data repository.

When looking at the user load generated on the network when clients access the LDS environment, we have to distinguish between the load being generated between the user and the Discovery Server—and the load generated between the portal, such as K-station.

When users access a Lotus Discovery Server environment, they do it with the K-map UI, which is locally downloaded to their machine and kept in the directory \WINDOWS\Downloaded Program Files to make loading and executing the K-map UI faster. All traffic they generate with this client on your network consists of the submission of the queries to the Discovery Server and the results they retrieve from the servers.

All this traffic connects to the primary server, no matter what portal server they are using when clicking the “Find”-link. Since the K-map is already usable at satisfying speeds when connecting to the K-map via 128 kbit/s ISDN lines, we recommend that you have users access the K-map server remotely—rather than having the servers access each other remotely through a slow WAN connection.

The reason for this recommendation is because the network traffic that two Discovery Servers produce is usually much higher than the network traffic that users can produce. Of course, there is a limit to this recommendation: when the number of concurrent remote users of the K-map exceeds a certain level, you should consider deploying another Lotus Discovery Server environment for that location.
Server recovery and troubleshooting

When problems arise, the pressure to get the server back up and running can be a source of conflict for the Discovery Server administrator who, on the one hand, needs to get the server back up and running, and on the other, wants to capture important diagnostic information to allow full resolution of the problem.

To strike a balance between the two sources of conflict, administrators need to have a clear understanding of how the various components of the Lotus Discovery Server work together. We recommend that administrators take the time to read Chapter 3, “Components and architecture” on page 37, which explains the server architecture in detail.

We also encourage all personnel involved in the administration and management of the Lotus Discovery Server to become familiar with the basic concepts and architecture of a Lotus Discovery Server environment. With such a foundation in place, administrators, managers, taxonomy editors, IT support staff, and so on will be able to communicate on a common understanding of how the Lotus Discovery Server has been deployed and be able to more effectively contribute to diagnosis and troubleshooting server problems.
7.1 Important - read the release notes first

We strongly recommend that all administrators read the online release notes which are installed as part of the Lotus Discovery Server installation. The file is a called readme.chm and is installed under the \lotus\kds\data\help folder.

The readme file is saved as a compiled HTML help file which can be launched by double-clicking it in Windows Explorer. It contains critical release notes you need to consider as part of your Lotus Discovery Server deployment. Inexperienced system administrators may find the Lotus Discovery Server to be a technically demanding product to install and configure, and are strongly advised to allocate plenty of time to read all the online documentation and generally familiarize themselves with the way the Lotus Discovery Server works.

Experienced administrators are also advised to resist the temptation to launch straight into the installation without reading the online documentation. For a full listing of recommended reading material and information sources, refer to 7.8, “Looking for answers” on page 231. Finally, if you haven't done so already, we recommend you read Chapter 2, “Planning a Lotus Discovery Server deployment” on page 13.

7.2 Why you need a server recovery plan

Dealing with server problems is best managed if the Lotus Discovery Server administrator can adopt a calculated, methodical approach which is based on sound problem management practices. To this end, one of the key themes emphasized throughout this chapter is the value of being able to refer to a Server Recovery Plan (SRP) which provides operational guidelines for dealing with IT infrastructure problems.

Well-prepared administrators usually have a set of procedures or a checklist in place which promotes efficient resolution of problems during the server recovery process. These procedures should be consistent with the guidelines laid down in the Server Recovery Plan. In this way, administrators can effectively initiate appropriate responses and actions in the event of a server failure.

Having access to Server Recovery Plan can also be beneficial in situations where:

- An experienced administrator is unavailable or leaves the company.
- Server recovery is being handled by an inexperienced administrator.
- New administrators are being trained on server recovery procedures.
Listed here are some broad areas of server operation your system administrators would be expected to deal with in the event of a server outage and/or problem. Your Server Recovery Plan should provide a framework which outlines what actions and responses are appropriate actions for managing problems in your IT infrastructure.

- Dealing with conflicts of interest
- Meeting Service Level Agreements (SLAs) specified for various components in your IT environment
- Managing server outages
- Recording/reporting server outages
- Monitoring and maintaining server performance
- Problem escalation

As part of your Lotus Discovery Server deployment, remember to update your response plan to accommodate specific requirements for monitoring, maintaining and troubleshooting your server infrastructure.

### 7.3 Dealing with conflicts of interest

When a server problem is reported, the Lotus Discovery Server administrator has to initially make a quick assessment of:

- What is the impact on users?
- How serious is the problem?
- What SLAs apply in this situation?

Figuring out the impact on users can be a tricky problem for the Lotus Discovery Server administrator. On a Lotus Discovery Server, there are no readily visible statistics the administrator can view on the server console. Domino administrators who are familiar with issuing the `show users` command at the Domino server console will find that the same command on a Lotus Discovery Server console will not list active users because HTTP connections do not create persistent sessions.

Metrics reports, however, can be generated to provide some post-analysis information on who is using the system. These reports can show you who is using the K-map and Search. By looking at these reports, the administrator can see who has been active on the system.

In any failure, the first part of the system to check out is whether the K-map is still available for user access. Loss of access to the K-map is a serious problem because users will not be able to:

- Submit searches
- View or navigate around the K-map
Locate expertise
View affinities
Edit the K-map

If the K-map is unavailable, it probably means there is a serious problem on the primary server. This in turn affects all secondary servers because most processes running on secondary servers rely on communicating with the primary XML to work queues. Checking out the status of the K-map is straightforward. Try submitting search or using the K-map Editor to open the K-map.

In terms of the impact on users, loss of the primary server is the most serious failure an administrator will have to deal with. If SLAs state your Lotus Discovery Servers should be recovered after 30 minutes (anything less would be difficult to achieve), the administrator will be under severe pressure to collect all important crash information and get the server back up within this time.

Loss of a secondary server is unlikely to have a direct impact on users and is more of an operational issue for the Lotus Discovery Server administrator. In this case, a different set of SLAs should apply. Knowledge of how various Discovery Services are distributed across secondary servers is essential. For example, K-map Indexing services user searches, whereas spidering a data repository is a background task that has no direct impact on user activity.

Service Level Agreements (SLAs) are a constant reminder to administrators that the system exists for the benefit of users. In negotiating a reasonable set of Service Level Agreements for your Lotus Discovery Servers, consider the following points:

- A Discovery Server outage is not a total loss of service for users.
- Not all users in your organization will access the Lotus Discovery Server environment and of these, an even smaller percentage of the user base are active at any point in time.
- Loss of a secondary server may have no direct impact on users.
- Loss of a primary server will probably have an impact on users.

On occasion, administrators may need to override SLA requirements to allow capture of critical system information for problem resolution purposes. In this case, short-term commitments to get the system up and running for user access are waived in favor of attaining long-term system stability through detailed investigation of the problem. In serious cases, the administrator may, under instruction from the vendor, collect important system crash information to help diagnose the problem.
### 7.4 Types of crashes

For organizations that have a Server Recovery Plan in place, guidelines for managing server problems will help administrators draw up appropriate procedures or checklists for recovering a server. For example, administrators will be expected to recover a server within pre-negotiated SLAs, log any server outages, and escalate serious problems to designated authorities or managers. The amount of pressure an administrator is under will depend to some extent on how well prepared they are for dealing with each type of server crash.

In the following sections, we describe three types of server failure the administrator is likely to encounter. Included under each failure type are symptoms you might see, what problems the user sees, and suggested responses the administrator might adopt. For all three failure types, the administrator must strike a balance between loss of service for users and the time required to diagnose, collect crash information, and recover the system.

#### 7.4.1 Degraded service

In a degraded service type of failure, a user still has access all parts of the system but experiences poor response times when accessing the server (for example, the K-map takes a long time to open). Table 7-1 lists symptoms and appropriate responses.

**Table 7-1  Degraded service**

<table>
<thead>
<tr>
<th>Server symptoms</th>
<th>Administrator response</th>
</tr>
</thead>
<tbody>
<tr>
<td>A degraded service is typically a result of poor system performance when server resources are unable to service demand.</td>
<td>Administrator has opportunity to collect critical system information while system is still running. Pressure to shut down and restart server depends on how slow performance is affecting users. Important evidence can be captured while users still have access to system.</td>
</tr>
<tr>
<td>This usually caused by insufficient memory, memory leaks, disk I/O bottlenecks, excessive processor demand, over-committed network utilization, or incorrect configuration of the system.</td>
<td>Examine system performance using tools like Windows Performance Monitor, Windows Task Manager or third-party performance tools. If necessary, upgrade hardware based on analysis of performance monitoring data.</td>
</tr>
</tbody>
</table>
7.4.2 Partial failure

In a partial failure, a user reports a problem with accessing certain parts of the system (for example, Sametime awareness not working properly). Table 7-2 lists symptoms and appropriate responses.

Table 7-2 Partial failure

<table>
<thead>
<tr>
<th>Server symptoms</th>
<th>Administrator response</th>
</tr>
</thead>
<tbody>
<tr>
<td>One or more tasks have generated an exception or abnormal termination event which renders the task(s) incapable of servicing or processing data requests.</td>
<td>Opportunities for collecting critical system information are reduced in this case because users may not be able to tolerate loss of functionality (e.g. can't access K-map). There is greater pressure to stop and restart the Lotus Discovery Servers. Response very much depends on what part of the system has failed.</td>
</tr>
<tr>
<td>Severity of this crash will depend on which tasks have stopped and how dependent users are on the tasks.</td>
<td>Any system failure on the primary server is probably intolerable from both a user and a system administration perspective. Failure on a secondary server such as spidering a data repository or updating the K-map Index could be tolerated pending an overnight reboot of the server. Where users are not directly accessing a secondary server, an immediate reboot of the is possible.</td>
</tr>
</tbody>
</table>

7.4.3 Complete failure

In a complete failure, a user reports the system is unavailable or not responding. Table 7-3 lists symptoms and appropriate responses.

Table 7-3 Complete failure

<table>
<thead>
<tr>
<th>Server symptoms</th>
<th>Administrator response</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lotus Discovery Server no longer responding. Task Manager shows one or more tasks as no longer responding. Server console possibly hung. Discovery Control Center not responding.</td>
<td>Administrator is under extreme pressure from users to get the system back up and running as soon as possible. Opportunities for collecting critical system information are under severe time constraints. Administrator needs to efficiently capture minimum amount of crash information as required in Server Recovery and Troubleshooting response plan.</td>
</tr>
</tbody>
</table>
7.5 Using Windows tools to diagnose a problem

Windows NT and Windows 2000 both have tools called Task Manager and Performance monitoring installed as part of the original operating system. These tools, in the hands of a skilled administrator, can be used to effectively monitor and diagnose most problems that may occur on a Lotus Discovery Server.

Administrators should at least be familiar with using Task Manager to examine resource usage and processes running on the server. Advanced administrators will be able to use the Performance monitoring tool to display graphs of system performance and generate logs for analyzing system performance.

We recommend that administrators compile system performance benchmarks for each Lotus Discovery Server to establish a baseline or reference point of normal server operation. Compiling such a set of benchmarks puts you in a strong position to analyze, monitor and even predict how changes in your system impact server performance. For example, predicting the impact of spidering a new data repository is possible if performance data has been previously captured for a similar data repository.

7.5.1 Windows Task Manager

Windows Task Manager can be used to capture real time resource usage on your Lotus Discovery Server. After loading Task Manager and selecting the Processes tab, we recommend you add the following performance counters to the default set of columns displayed:

- Virtual Memory Size
- I/O Reads
- I/O Writes
- I/O Read Bytes
- I/O Write Bytes

Note: You may need to resize the Task Manager window to see all columns in one screen.

Figure 7-1 shows how the Task Manager can be used to display useful information on various processes running on your Lotus Discovery Server. This can be achieved with very little effort by the administrator.
In looking at the processes running on this example server, three points of interest can be found:

1. The db2syscs.exe process is using a relatively large amount of virtual memory.
2. The ncmserv.exe process performs numerous I/O Reads which fetch small amounts of data per read.
3. The natg.exe process reads significantly more bytes than other tasks and performs fewer I/O Reads than the ncmserv.exe process.

Switching to the Performance tab within Task Manager can provide useful information on CPU and memory usage.
Looking at this tab, you can immediately see total memory usage exceeds the total physical RAM installed on the server. Your initial reaction might be to add more physical RAM to avoid excessive memory paging. However, beware of jumping to a quick conclusion. On closer examination of the graph, pay attention to the Physical Memory box and see how much RAM is still available. If you refer back to Figure 7-1 on page 212, which lists processes running on the server, you’ll find the DB2syscs.exe process is consuming both physical and virtual memory.

The point to keep in mind here is that basic tools like Task Manager and the Performance monitor can provide, at little cost, a lot of valuable information on how your system is performing.

7.5.2 Windows Performance monitor

To examine server processes on a timeline basis, the Windows Performance monitoring tool can be used to produce graphical and/or log output. Logged output can be saved in a variety of file formats and analyzed later using spreadsheet tools and statistical analysis packages. For more information on how to use the Performance monitoring tool to analyze server performance, refer to 6.4.4, “Monitoring performance” on page 196.

7.6 Capturing crash information

It is the job of the administrator to collect as much useful information as possible in the event of a server failure. The ability to collect crash information will be depend on a number of factors, most of which are the control of the Lotus Discovery Server administrator.

As mentioned in 7.3, “Dealing with conflicts of interest” on page 207, the administrator is under considerable pressure to get the system back up and running. New or inexperienced administrators should familiarize themselves with response guidelines specified in the Server Recovery Plan to ensure that appropriate procedures have been followed to recover the server.

Remember, the server recovery process doesn’t end when the server is back up and running. Instead, it includes recording, reporting, and post-analysis of crash information collected during the recovery process.
7.6.1 Don’t destroy the evidence

Rigorous recording of crash information and capture of evidence can be a time-consuming process. However, failure to record crash information can lead to repeated server crashes and prolonged instability for users because important evidence has been ignored or lost.

In some situations, the administrator may choose to deliberately delay shutting down and restarting a server to allow critical system information to be captured for deeper diagnosis of an ongoing problem. For example, evidence on how much memory a particular process is consuming may mean delaying a server shutdown to allow Task Manager and/or the Performance monitor to be used to examine resource usage.

Rebooting a server without first analyzing what is currently happening on the server can prove to be a frustrating exercise for all concerned. Critical diagnostic information can be lost and you may be setting yourself up for a lengthy periods of server instability.

7.6.2 Screen dumps, errors and logs

The importance of accurately capturing crash information cannot be overstated. When writing down error messages from the server console, make sure the error message is recorded exactly as it is presented on the screen (including upper case/lower case characters). The ability of a support analyst to examine and successfully diagnose a problem reported by a customer relies heavily on the accuracy and completeness of the information submitted. Following are the sources of information you should consider collecting.

Screen dumps

Screen dumps are one of the most effective ways of accurately capturing the current status of your system. On most servers, you can press one or two keys to dump screen information to the clipboard (refer to Table 7-4).

<table>
<thead>
<tr>
<th>Keys</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Print Screen</td>
<td>Captures the entire screen</td>
</tr>
<tr>
<td>ALT + Print Screen</td>
<td>Captures the currently active window</td>
</tr>
</tbody>
</table>

* Actual Print Screen Keys may vary depending on your system.

Initially the screen output is saved to the system clipboard. From there you can paste the contents of the clipboard into Window Paint and save the output as an image file.
The Lotus Discovery Server console

The contents of the Lotus Discovery Server console can be captured by marking and copying the contents of the console window (see Figure 7-2). If you run the server console in full screen mode, press the ALT and Enter keys together to switch to window mode. The copied information is saved to the clipboard as text. From there you can paste the contents of the clipboard into Notepad and save the output as a text file.

![Figure 7-2 Marking the contents of the server console](image)

**Tip:** You can increase the amount of information stored in the Lotus Knowledge Discovery System console by increasing the height of the screen buffer size - see Figure 7-3 on page 216.
Inside the Lotus Discovery Server

Figure 7-3   Increasing the screen buffer size

Windows dialog boxes
Error messages displayed in dialog boxes can be easily captured to a file. First, switch focus to the dialog box by clicking the Title Bar. Now dump the contents of the active window to the clipboard by pressing the ALT and Print Screen keys together. Now save to a file using Windows Paint.

Task Manager
Task Manager can be loaded to view system resource usage and process information. This tool can be used to view the current status of your Lotus Discovery Server and diagnose resource usage problems (e.g. running out of memory). Information displayed in the Task Manager window can quickly be dumped to the clipboard by pressing the ALT and Print Screen keys together.

Log output
The Lotus Discovery Server writes log output to a file called discoverylog.nsf. To accurately capture server activity, consider making a copy of this file to send to your support analyst. Be careful, though; this file can be very large.

Tip: To manage the size of the discoverylog.nsf file, write a simple agent which selectively purges documents based on your criteria (e.g. older than 90 days).

Server configuration settings
The Lotus Discovery Server server document contains configuration settings which control access to the server, security settings, servlets, http access, etc. Information in this document can easily be cut and pasted into a file to send to a support analyst.
Dashboard.nsf
The dashboard.nsf database stores a great deal of useful information about how you have configured your Lotus Discovery Server system. A support analyst can examine this file to check configuration settings and see what data repositories you are spidering.

XML queues
Your support analyst may ask you to examine various XML queues to determine the status of your system. Using Windows Explorer, you can list which work queues reside on your servers. Displaying the file size and last modified date/time can reveal important information about the current status of your system (see Figure 7-4 on page 221 as an example of this). If necessary, you can screen dump the contents of Windows Explorer to the clipboard and save the output to a file.

Note: All XML work queues are initialized as a 4 KB file.

7.6.3 Recording incidents
Guidelines for determining what information should be recorded in the event of a system failure should be drawn from your Server Recovery Plan. Listed here is an example of the type of information we suggest you record in a server incident report. Ideally this information should be entered into a database (e.g. Notes) for easier searching, viewing, and analysis of server problems.

<table>
<thead>
<tr>
<th>Reported by:</th>
<th>Andy Crashman</th>
</tr>
</thead>
<tbody>
<tr>
<td>Server name:</td>
<td>LDS01/Atlanta</td>
</tr>
<tr>
<td>Date of incident:</td>
<td>04/13/2001</td>
</tr>
<tr>
<td>Time incident reported:</td>
<td>10:15am</td>
</tr>
<tr>
<td>Time incident resolved:</td>
<td>10:35am</td>
</tr>
<tr>
<td>Status:</td>
<td>Resolved</td>
</tr>
<tr>
<td>Crash symptoms:</td>
<td>Discovery Control Center reports Notes Spider no longer responding. Started spidering at 8:30am. Lotus Discovery Server still running and able to respond to console commands. XML queues on Primary Discovery Server not receiving any new data</td>
</tr>
</tbody>
</table>
We recommend establishing some form of incident reporting system which logs all problems found on your servers. Maintaining an audit log of all server incidents can provide valuable context for ongoing server problems which, in isolation, provide no clues as to why the server crashed. Consider the following scenario which illustrates this point.

**Example scenario**
You are investigating a problem where one of your Lotus Discovery Servers crashes after nearly 200 hours of continuous operation. The problem has occurred five times over the last two months. In looking back through the incident reports, you notice another incident report for the same server is always raised the day before the Lotus Discovery Server crashes. It states the weekly backup of the server failed.

Here is an example of how context has been added to the problem. A pattern starts to emerge where Lotus Discovery Server crashes are preceded by a weekly backup failure.

Further investigation reveals the Lotus Discovery Server is always rebooted after a successful completion of a weekly backup. Further testing reveals a memory leak in introduced on the Lotus Discovery Server after the weekly backup fails. 24 hours later, the Lotus Discovery Server crashes when all available memory is consumed.

<table>
<thead>
<tr>
<th>Error message(s):</th>
<th>Spider log reports “Error - unknown or corrupt attachment”</th>
</tr>
</thead>
<tbody>
<tr>
<td>(include screen dumps here)</td>
<td>Action taken: Removed problem document from database and rescheduled spider. Database now spiders okay. Now suspect “Distiller” program is generating corrupt .pdf files which cannot be opened by the Keyview filter.</td>
</tr>
<tr>
<td><strong>Synopsis of the problem:</strong></td>
<td>Spider unable to process corrupt .PDF attachment.</td>
</tr>
<tr>
<td><strong>Other information:</strong></td>
<td>Previous incident reports show third time this week. Always stops when processing documents with a .PDF file that was created programatically by software called “Distiller”. Acrobat Reader can’t open attachment.</td>
</tr>
<tr>
<td><strong>Resolution:</strong></td>
<td>Programmer will modify “distiller” program to produce correctly formed .PDF files.</td>
</tr>
</tbody>
</table>
In this example, the root cause of the problem could have easily remained undetected if the problem had not been viewed in context. The administrator could have (incorrectly) concluded the Lotus Discovery Server had a slow memory leak which could be “managed” by rebooting the server every Sunday night (this would only work if the weekly backup was run earlier in the day). By examining a complete log of incident reports for the server, an association between failed backups and Lotus Discovery Server crashes could be made.

7.6.4 Location of Lotus Discovery Server system files

In developing a server recovery response plan for your Lotus Discovery Servers, it is a good idea to compile a checklist or table of where important system information is stored. This can then be used by the administrator to efficiently locate and capture relevant information in the event of a system failure. Table 7-5 lists most of system files that are installed as part of a standard Lotus Discovery Server implementation.

Note: In a distributed implementation of the Lotus Discovery Server, some of these files may be located on the secondary rather than the primary Discovery Server.

<table>
<thead>
<tr>
<th>Type</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>Notes Databases</td>
<td>\lotus\kds\data\dashboard.nsf  \lotus\kds\data\discoveryadmin.nsf \lotus\kds\data\discoveryserver.nsf \lotus\kds\data\indexmap.nsf \lotus\kds\data\people.nsf \lotus\kds\data\peopleq.nsf</td>
</tr>
<tr>
<td>Logs</td>
<td>\lotus\kds\data\log.nsf  \lotus\kds\data\discoverylog.nsf \lotus\kds\db2log.log \lotus\kds\data\cmap.log \lotus\kds\data\metrics.log \lotus\kds\data\db\db2\db2diag.log</td>
</tr>
<tr>
<td>.ini</td>
<td>\lotus\kds\Notes.ini  \lotus\kds\formats.ini \lotus\kds\keyview.ini</td>
</tr>
</tbody>
</table>

Table 7-5 Location of key Lotus Discovery Server files
<table>
<thead>
<tr>
<th>Type</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>.chm</td>
<td>\lotus\kds\data\help\install.chm</td>
</tr>
<tr>
<td></td>
<td>\lotus\kds\data\help\readme.chm</td>
</tr>
<tr>
<td></td>
<td>\lotus\kds\data\help\ds_admin.chm</td>
</tr>
<tr>
<td></td>
<td>\lotus\kds\data\help\kds_user.chm</td>
</tr>
<tr>
<td>XML work queues (see Figure 7-4 on page 221)</td>
<td>\lotus\kds\data\</td>
</tr>
<tr>
<td>DB2 databases</td>
<td>\DB2*.*</td>
</tr>
<tr>
<td></td>
<td>\DB2CTLSV*.*</td>
</tr>
<tr>
<td>QuickPlace databases</td>
<td>\lotus\kds\data\quickplace\quickplace</td>
</tr>
<tr>
<td>K-station files</td>
<td>\lotus\kds\data\p\</td>
</tr>
<tr>
<td>Domino Servlets</td>
<td>\lotus\kds\data\domino\servlet</td>
</tr>
</tbody>
</table>
7.7 Top ten configuration problems

The following “top ten” problems have been compiled from problem reports and several discussion forums managed by Lotus. They have been selected on the basis of providing users with information that will maximize the chances of successful deployment of a Lotus Discovery Server solution for your organization.
7.7.1 Common name = computer name = host name

Sounds simple enough, this problem really caused users a lot of grief when setting up their Lotus Discovery Servers. Much of the confusion results from misinterpretation of terminology used in Domino and Windows.

First, let's define each of the three components.

**Computer name**

The computer name is what you called your Windows NT/2000 server when you set it up. It does not include the DNS suffix. In Figure 7-5, the computer name is “speckledhen”.

![C:\ipconfig /all](image)

*Figure 7-5  Computer name is “speckledhen”*

**Common name**

The online installation documentation states you must register your KDS Server name in your existing Domino domain using a name that exactly matches your WinNT machine name. Well, this is not quite true.

What you really need to do is register a Lotus Discovery Server with a “common name” that exactly matches your computer name (as defined above). The key point here is “common name”. The safest and clearest way to describe common name is by example, so let's start with a Lotus Discovery Server which has been registered with a hierarchical name of “speckledhen/srv/demo”. In this case:

- “speckledhen” is the common name
- “srv” is the organizational unit
- “demo” is the organization name

So, if your Windows NT/2000 computer name is “speckledhen”, you could register any of the following Lotus Discovery Server names:
The issue of what organizational unit (if any) and organization name you use is only important from a Domino server perspective. In a production environment you will probably register the Lotus Discovery Server in an existing Domino domain, and existing rules for organizational units and organization will apply.

In our demonstration environment, we have registered a server called "speckledhen/demo"; refer to Figure 7-6.

![Figure 7-6 Registered (Domino) Lotus Discovery Server name](image)

**Host name**

The host name is defined here as the left-most part of the fully qualified Internet host name. Again, this is best dealt with by example.

Let's take a fully qualified Internet host name of "speckledhen.lotus.com". If we only take the first part of the name up to the first ".", we get "speckledhen". This is the host name and it matches our computer name and our Lotus Discovery Server common name.

**Tip:** To verify the host name is responding okay, try pinging it from a Command Prompt window.

**Watch out for DNS**

DNS is another source of confusion for many administrators. The main issue to confirm before you set up your Lotus Discovery Servers is whether you can ping the host name from another computer. For example, Figure 7-7 on page 224 shows the result of pinging a host name of speckledhen:
If you cannot successfully ping the host name, chances are the entry is not defined in your organization's DNS table(s). As an interim measure, you can create a local host table entry to resolve your host name to an IP address. Figure 7-8 shows how to do this:

```
C:\>ping speckledhen
Pinging speckledhen.lotus.com [9.95.35.68] with 32 bytes
Reply from 9.95.35.68: bytes=32 time<10ms TTL=127
Reply from 9.95.35.68: bytes=32 time<10ms TTL=127
Reply from 9.95.35.68: bytes=32 time<10ms TTL=127
Reply from 9.95.35.68: bytes=32 time<10ms TTL=127
Ping statistics for 9.95.35.68:
   Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
   Approximate round trip times in milli-seconds:
      Minimum = 0ms, Maximum = 0ms, Average = 0ms
C:\>
```

Figure 7-7  Pinging a host name

Figure 7-8  Adding a host name to your local host table
Chapter 7. Server recovery and troubleshooting

Summary of connectivity
Table 7-6 summarizes how your Lotus Discovery Server name, computer name and host name work together. By now you probably understand why two of the examples are okay and why the other two are wrong.

Table 7-6  Common name = computer name = host name

<table>
<thead>
<tr>
<th>LDS hierarchical name</th>
<th>Computer name</th>
<th>Host name*</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>speckledhen/srv/atlanta</td>
<td>speckledhen</td>
<td>speckledhen</td>
<td>okay</td>
</tr>
<tr>
<td>speckled/svr/atlanta</td>
<td>speckledhen</td>
<td>speckledhen</td>
<td>wrong</td>
</tr>
<tr>
<td>speckledhen/altlanta</td>
<td>speckledhen</td>
<td>speckledhen</td>
<td>okay</td>
</tr>
<tr>
<td>speckledhen.xyz.com/atlanta</td>
<td>speckledhen</td>
<td>speckledhen</td>
<td>wrong</td>
</tr>
</tbody>
</table>

* Fully qualified Internet host name in this example is “speckledhen.xyz.com”

7.7.2 Why don’t I have any affinities

The first area to check is your affinity threshold. In the Lotus Discovery Control Center, go to step “5. Generate Affinities” and select “Specify Settings”. Make sure you have not forgotten to change the default setting of 100% to something suitable for your site. The setting is deliberately defaulted to 100% so you do not send proposed affinity e-mails to users until you are ready.

Affinity Generation Threshold
The Affinity Generation Threshold is the point when a person’s knowledge about a K-map category becomes significant compared to others.

A 50-60% threshold will likely be high enough to filter out weaker people-to-category relationships, but not so high as to never propose any affinities

Threshold: 100 %

Figure 7-9  Setting Affinity threshold

7.7.3 I can’t view/open my Lotus Discovery Server logs

All log activity is stored in a file called discoverylog.nsf. This file exists on all Lotus Discovery Servers, but is only active on the primary server.
If you are experiencing problems viewing log output in the Lotus Discovery Control Center, go to Windows Explorer and check out the size of the log file (refer to Figure 7-10). If the size is excessively large, you either have a problem with:

- Too many records, because a Discovery Server process is generating excessive output. Web spiders are a good place to start your investigation, as these can generate a large number of warning messages.

- Keeping too many active records. Consider writing a simple Notes agent that archives records older than 90 days.

![Figure 7-10 Checking out the size of the discovery server log file.](image)

### 7.7.4 Spidering a Web site never seems to complete

Spidering a Web site can be an extremely lengthy process if you select a large Web site and don’t apply restrictions on how you configure the spider.

We recommend you adopt a very conservative approach when spidering a Web site for the first time. By “conservative”, we mean configuring your Web spider to not follow links to other sites and limiting the number of levels you spider. In Figure 7-11, we have chosen to not follow links to other servers and we only process pages four levels deep.

![Figure 7-11 Web spider settings](image)

**Tip:** To avoid flooding your production environment with large amounts of newly spidered Web documents, we suggest doing a test run using a test or “staging” environment.
7.7.5 I can’t see any categories in my K-map

The K-map Editor won’t display any categories to review until you build the K-map. Before you can build the K-map, you must successfully spider some data repositories. Following is a step-by-step check you can follow, from initial spidering to K-map creation.

At the Lotus Discovery Control Center:

- Check the spider log for each data repository you spidered. Make sure the data repository was successfully spidered.
- Verify that you have enabled the K-map Indexing, Repository Spider and Metrics Data Collection tasks when you spider the Data Repositories.
- Verify that you have allowed plenty of time for the repository spidering and subsequent metrics collection processes to complete.

Once a data repository has been successfully spidered, it becomes eligible for inclusion in the initial set of data repositories you choose to build the K-map from. The K-map (taxonomy) creation process requires careful selection of one or more data repositories. Verify that you have data repositories with:

- Data that is representative of the problem you want to solve
- Enough documents to build a K-map
- Documents with enough rich content to qualify for inclusion in the K-map

When you select the repositories to create your initial K-map, you enable the K-map building task. Under step “3. Create K-map”, select “Monitor Creation” to open the K-map Building log. The log will tell you:

- How many documents were used to create the K-map. For example:
- How many documents were too small. For example:
  - 872 documents too small
- Whether K-map creation completed successfully. For example:
  - 04/13/2001 11:26:28 PM K-map creation complete using 1140 documents

If all of these steps have been successfully completed, you should have an initial K-map to review and edit.

One final area to check is your K-map settings that control the K-map creation and categorization rules. We recommend you stick with the default settings unless you think they are unreasonable for your situation.
7.7.6 Cannot administer QuickPlace from Administration Place

If you are trying to administer QuickPlace from the QuickPlace page in K-station, and you are unable to authenticate with the realm quickplace/quickplace, you will receive the error message: Warning unable to sign into this quickplace.

The most likely reason for this error is you did not add at least one individual member to the KDSAdministrators group. During installation of the server, you have the opportunity to specify additional individual users in the KDS Administrators field (see Figure 7-12). Remember to separate multiple entries with a comma or semicolon.

Specify the Domino users(s)/group(s) who will administer Knowledge Directory

<table>
<thead>
<tr>
<th>KDS Administrators:</th>
<th>KDSAdministrators, John Doe</th>
</tr>
</thead>
</table>

Figure 7-12 Specify individual names in KDSAdministrators group

If you do not specify any individual user names, either explicitly or via the KDSAdministrators group, no one will be authorized to administer the QuickPlace server because all four QuickPlace databases will have an incorrect ACL entry that's missing your Domino domain name.

Workaround

1. Shut down the Lotus Discovery Server.
2. On another machine, open the Notes client using the ID of someone in the KDSAdministrators group.
3. In the Access Control Lists of the four databases in the quickplace/quickplace directory on the server (Admin.nsf, Contacts1.nsf, CreateHaiku.nsf, and Main.nsf), make sure the group KDSAdministrators includes the organization name (and country code, if you're using one). For example:
   - KDS/Administrators/quickplace/<organization> or
   - KDS/Administrators/quickplace/<organization>/<countrycode>
4. Restart the server.

Tip: To reduce the amount of work involved in editing your K-map, try reduce the maximum number of sub-category levels in K-map (depth) to two.
7.7.7 Server security settings are incorrectly set

The server document contains a number of security settings which control access to Java applets, servlets, lotusscript and HTTP browsing. Instead of trying to describe a particular problem that is triggered by one of these settings, refer to 5.3.1, “Server security” on page 154. Here you will find a detailed explanation of how to set up your security settings in the server document. If you do this correctly, you'll eliminate many errors commonly reported for both the Lotus Discovery Server and K-station.

7.7.8 KDSAdministrators group is missing important entries

The KDSAdministrators group is an important group which is used in database ACLs and in several places in the Server Document to allow privileged access to all parts the Knowledge Discovery System (i.e. Lotus Discovery Server and K-station). Figure 7-13 shows an example of the contents of the KDSAdministrators group we used to set up our Demonstration server.

![GROUP: KDSAdministrators](image)

At minimum, the KDSAdministrators group should contain:

- The primary Lotus Discovery Server name. For example:
  - speckledhen/demo
- The “KDS/Lotus Notes Companion Products” entry
- At least one individual hierarchical user name. For example:
  - Dom Admin/Demo
7.7.9 Access to the Domino Directory

This is a security issue of who is allowed to modify the ACL of the Lotus Domino Directory in the domain where the Lotus Discovery Server has been registered. During installation of the Lotus Discovery Server, the installation program attempts to automatically create (or if they already exist, edit) several groups in the Domino Directory. To enable this, you need to at least add the entry “KDS/Lotus Notes Companion Products” with either:

- Author access (with group creator/modifier roles) to create new groups
- Editor access (with group creator/modifier roles) to update existing groups

Communicate with your IT security contact early on to ensure that access to the Domino Directory does not become an issue during installation and setup of the Lotus Discovery Server.

Workaround

Before installing your first Lotus Discovery Server in the Domino domain, get your Domino administrator to create the groups listed in Table 7-7 in the Domino Directory. Next, populate each of these groups with the appropriate members.

Table 7-7 Special groups created in the Domino Directory

<table>
<thead>
<tr>
<th>Group name</th>
<th>Sample members</th>
</tr>
</thead>
<tbody>
<tr>
<td>KDSAdministrators</td>
<td>Dom Admin/USR/Demo</td>
</tr>
<tr>
<td>(Individuals authorized to initially configure and maintain all KDS services)</td>
<td>KDS/Lotus Notes Companion Products</td>
</tr>
<tr>
<td></td>
<td>Speckledhen/SRV/Demo</td>
</tr>
<tr>
<td></td>
<td>Bass/SRV/Demo</td>
</tr>
<tr>
<td>AffinityDesignators</td>
<td>Cathy Powers/USR/Demo</td>
</tr>
<tr>
<td>(Usually supervisors or managers who are knowledgeable about the Category content in the K-map. Empowered to recommend affinity associations with specific categories in the K-map.)</td>
<td>KDSAdministrators</td>
</tr>
<tr>
<td>DiscoveryServers</td>
<td>Speckledhen/SRV/Demo</td>
</tr>
<tr>
<td></td>
<td>Bass/SRV/Demo</td>
</tr>
</tbody>
</table>
7.7.10 First-time start up error messages on LDS server console

You might see one, or both, of the following messages at the server console the first time you start up the Lotus Discovery Server:

- Unable to execute command (load dssched) on Domino Server: You are not authorized to use the remote console on this server.

  **Workaround**
  
  Restarting the Lotus Discovery Server resolves the problem - assuming that the Server document in the Domino Directory is configured properly. Verify that the group DiscoveryServers is entered in the Administrator field in the Basics tab.

- AMgr: Agent 'ConflictHandler' in 'lp\lpcat.nsf' does not have proper execution access, cannot be run.

  **Workaround**
  
  Restarting the Lotus Discovery Server resolves the problem - assuming that the Lotus Discovery Server document in the Domino Directory is configured properly.

7.8 Looking for answers

There are a variety of information sources you can easily access to look for answers to your questions. There is online documentation available on the Lotus Discovery Server, Web discussion forums, and the usual support resources like the Lotus Knowledge Base and Lotus Customer Support.
7.8.1 Online documentation is context-sensitive

The online documentation installed with the Lotus Discovery Server is context-sensitive. This applies both in the initial setup and configuration of the Lotus Discovery Server, and in the Discovery Server Control Center. A help button is always located somewhere near the top right-hand corner of the screen you are currently in (refer to Figure 7-14). When the relevant section from the help file is loaded, you can still navigate to any other part of the help documentation as required.

![Lotus Knowledge Discovery System Setup](image)

**NOTE:** You must register this server in an existing Domino domain before proceeding. Refer to the ReadMe for more details, and to see the Release Notes.

### New Server Settings

**Server Type:**
- **Primary:** The primary server manages and maintains the master Knowledge Discovery System databases, and sends them to their replicas on any secondary servers. Only one primary server can exist per Knowledge Discovery System distribution.
- **Secondary:** Secondary servers receive replica copies of Knowledge Discovery System databases from the primary.

![Image of Lotus Knowledge Discovery System Setup with Help button highlighted](image)

**Figure 7-14 Access to context-sensitive help**

7.8.2 Compiled HTML help files

Compiled HTML help files are installed on every Lotus Discovery Server under the \lotus\kds\data\help folder. Table 7-8 lists the file names and titles of these help files.

<table>
<thead>
<tr>
<th>File name</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>install.chm</td>
<td>Lotus KDS/K-station Install and Setup Help</td>
</tr>
<tr>
<td>readme.chm</td>
<td>Lotus KDS 1.0 and K-station 1.0a Readme File</td>
</tr>
</tbody>
</table>
7.8.3 Notes.net discussion forum

Lotus has established a Knowledge Discovery System (KDS) discussion forum which covers both Lotus K-station and Lotus Discovery Server. The URL is:

http://www.notes.net/kdsforum.nsf

Lotus has made this forum available for you to continue to provide feedback and ask questions of Lotus and your fellow customers and business partners. You are encouraged to keep the communication lines open and use this forum early and often, as your input will help us ship better products. You can also use the forum to keep abreast of announcements and other reference materials about K-station and the Discovery Server.

7.8.4 Notes.net documentation library

The Documentation Library features documentation and online Help for most Lotus products. The initial URL is:

http://www.notes.net/doc

From there, select the link Knowledge Discovery System User Help, Installation Help, and more to read the documentation online, or download the files for printing or viewing.
Planning a backup strategy for KDS

Existing backup strategies, which have been developed for Domino servers, will need to be modified to cover additional software components that are installed as part of a KDS installation. The main difference you need to allow for is backup of the DB2-based K-map that is installed on the primary Lotus Discovery Server.

For a distributed server implementation, the backup strategy is complicated by the fact that you have to identify what Discovery Services have been installed on your secondary server(s). You also have to decide whether you want to perform full system backups of your servers (the simplest option) or selective backups to minimize downtime and the storage costs of your backup media.

A key issue to keep in mind when developing your backup strategy is the amount of time the primary Lotus Discovery Server will be down. This is important because secondary servers cannot tolerate loss of access to the primary server. The rule of thumb to apply in any strategy is “primary server down last and back up first”.
8.1 Developing a backup strategy

To establish what type of backup strategy is most suitable for your KDS infrastructure, try answering the following set of questions about the status of your current backup strategy and its capacity to meet your requirements for backing up your KDS servers. Your answers to these questions should help you identify issues that need to be addressed before you start deploying KDS as a strategic component in your IT environment.

- What backup software are you currently using?
- What does your current software license permit you to do?
- What backup hardware do you have available?
- What will be the down time policy for your KDS servers?
- What impact will the scheduled down time have on each of the Discovery Services (spidering, indexing, etc)?
- What backup, recovery and disaster recovery policies does your company currently have in place?
- How much data do you expect to back up and at what speed?
- What is the anticipated growth of your data?
- Can your software back up open files?

8.1.1 Backup architecture

For a distributed Lotus Discovery Server implementation, the following architecture diagrams are offered as examples of how you might design your backup solution. A full system backup or selective backup of files can be implemented for either solution.

**Basic backup solution to directly attached tape unit**

The first solution, shown in Figure 8-1 on page 237, proposes backing up each physical server to an attached tape device. Features of this solution are:

- Simplest backup strategy to deploy
- No additional network traffic
- Higher backup media costs
- Higher ongoing operational costs
- Per-server licensing of backup software
- Primary server needs highest speed, largest capacity backup hardware
Solution becomes difficult to operate when backups span multiple tapes especially if primary server is involved.

Network backup to a tape library
The second backup solution, shown in Figure 8-2 on page 238, represents a typical enterprise solution where all data is backed up to a centralized tape library. Features of this solution are:
- More complicated backup strategy to initially deploy, if first time
- Additional network traffic
- Lower backup media costs
- Lower ongoing operational costs
- Existing backup infrastructure may cover KDS infrastructure
- Backup infrastructure likely to be high speed and high capacity solution
- Solution able to handle multi-reel media
Mirror backup solution

The third backup solution, shown in Figure 8-3 on page 244, involves clustering or mirroring of data to another server. This solution allows you to minimize server downtime, because the mirrored images of your primary and secondary servers can be created in real time while the servers are up. The time-consuming task of backing up the mirrored servers can be done offline without impacting the production system. At a minimum, you need to mirror the primary server. The need to mirror your secondary servers will depend on what Discovery Services they are hosting.
8.2 Backing up distributed Discovery Services

For a distributed installation of KDS, you first need to identify which Discovery Services are installed on what secondary KDS server and then identify what files are associated with each Discovery Service. This is particularly important if you decide to perform selective (rather than full) backups of each server.

Table 8-1 lists backup requirements on a “per Discovery Service” basis, to help you understand the specific requirements of each component. By knowing what files are associated with each service, you will be able to work out what parts of each primary and secondary server need to be backed up.

<table>
<thead>
<tr>
<th>Discovery Service</th>
<th>Folder/File to back up</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>K-map Building (ATG)</td>
<td>ATGRestartQ.queue</td>
<td>Can be installed on a secondary server. All files and folders to be backed up will be found under the Lotus\KDS\data folder.</td>
</tr>
<tr>
<td></td>
<td>ATGoutofbounds.queue</td>
<td></td>
</tr>
<tr>
<td></td>
<td>All stopword*.txt files</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ATG_Root</td>
<td></td>
</tr>
<tr>
<td>K-map Indexing</td>
<td>FTDOMAIN.DI</td>
<td>Can be installed on a secondary server. All files and folders to be backed up will be found under the Lotus\KDS\data folder.</td>
</tr>
<tr>
<td></td>
<td>Indexmap.nsf</td>
<td></td>
</tr>
<tr>
<td></td>
<td>LDSFullTextRestart.00</td>
<td></td>
</tr>
<tr>
<td></td>
<td>through</td>
<td></td>
</tr>
<tr>
<td></td>
<td>LDSFullTextRestart.07</td>
<td></td>
</tr>
<tr>
<td>Metrics Processing</td>
<td>people.nsf</td>
<td>Can be installed on a secondary server. However, you should back up these files from the primary server, which maintains “master” replicas of them.</td>
</tr>
<tr>
<td></td>
<td>people.ft</td>
<td></td>
</tr>
<tr>
<td></td>
<td>peopleq.nsf</td>
<td></td>
</tr>
<tr>
<td>KDS server configuration</td>
<td>discoveryserver.nsf</td>
<td>You should back up these files from the primary server, which maintains “master” replicas of them.</td>
</tr>
<tr>
<td></td>
<td>discoverylog.nsf</td>
<td></td>
</tr>
<tr>
<td></td>
<td>dashboard.nsf</td>
<td></td>
</tr>
<tr>
<td>K-station</td>
<td>LP</td>
<td>Can be installed on a secondary server. All files and folders to be backed up will be found under the Lotus\KDS\data folder.</td>
</tr>
</tbody>
</table>
8.3 Backing up open files

One of the challenges in backing up Lotus Discovery Servers is handling open files. Because this situation is common with most database systems, many organizations have an existing solution to this problem. Options for handling open files are:

- Shut down all KDS services.
  If your backup software is not capable of backing up open Notes databases and DB2, you must shut down the server, back up your files, then restart the server. This provides an inexpensive and reliable backup solution for your system. However, for KDS systems, finding enough available time to shut the servers down can be a problem.

- Replicate Notes databases to another Domino server.
  The underlying Domino infrastructure used by KDS allows you to replicate Notes databases to another Domino server. This approach will cover some of your backup requirements, but still needs to be supplemented with a backup solution that backs up non-Domino files created by KDS.

- Purchase or create backup software that can handle open files.
  Several vendors offer advanced backup software that has the ability to detect when there are no pending write requests for a file. Remember, if you want to back up KDS Notes databases that have transaction logging enabled, you will need software that was written specifically for the Lotus C API for Domino and Notes.

- Replicate DB2 to another host.
  Technically, it is possible to replicate DB2 to another server. This feature may be offered in a feature release of KDS. Administrators who have expertise in this area may be able to develop their own backup solution for replicating DB2 to a backup server.

8.4 Stopping and restarting KDS services

If your backup strategy requires the KDS servers to be shut down, you will need to shut down your servers in a controlled, orderly sequence to reliably obtain a complete system backup. As stated in the introduction to this chapter, you must always shut down the primary server after all secondary servers have been shut down.
On restart, the reverse is true—the primary server must be completely up and running before you restart your secondary servers. This is because Discovery Services running on secondary servers rely heavily on reading and writing to the primary server. For example, all spider types (Web, File, Notes, Directory) always write XML output to the primary server.

### 8.4.1 Shutting down a Lotus Discovery Server

The first step in shutting down a Lotus Discovery Server is to issue the command `tell KDS shutdown` at the server console. After all KDS servers have been shut down, stop DB2 on the primary KDS server. To manually stop each of these services, load the Windows Services applet (under Control Panel), and for each service, apply the Stop command.

Table 8-2 lists all active DB2 services that need to stopped on the primary Lotus Discovery Server.

<table>
<thead>
<tr>
<th>DB2 service</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB2</td>
</tr>
<tr>
<td>DB2CTLSV</td>
</tr>
<tr>
<td>DB2DAS00</td>
</tr>
<tr>
<td>DB2JDBC Applet Server</td>
</tr>
<tr>
<td>DB2 License Server</td>
</tr>
<tr>
<td>DB2 Security Server</td>
</tr>
</tbody>
</table>

**Tip:** Create a script which first stops the KDS server, and then all DB2 services, before initiating a system backup. Upon completion of the backup, restart all DB2 services and then all KDS servers.

To test if the primary KDS server is available, try opening the Domino Directory using an API call.

### 8.5 K-map - key backup points

Your key backup points will change as you progress down the path of deploying your KDS infrastructure. During initial creation and building of the K-map, key points for backing up your KDS server(s) are:
After creating the K-map from a representative sampling of repositories, but before K-map editors relabel.

This allows you to recover to the initial machine-generated state of the K-map. Relabelling of machine-generated categories can be a demanding task for the uninitiated and inexperienced K-map editor. If the relabelling process produces unsatisfactory results, the original K-map can be recovered to a known consistent state and the K-map editors can apply what they have learned from their previous efforts.

- After K-map editors have relabeled but before enabling categorization.

If the categorization process produces unsatisfactory results, the K-map editors can recover their work-to-date and make adjustments to produce a better result.

- At least once a week, depending on the frequency of changes and your organization's backup policy.

8.6 Establishing a backup schedule

For many system administrators, the most appropriate strategy for backing up Lotus Discovery Servers will not become apparent until some experience has been gained on how the servers are performing and whether they have reached a steady state of operation. A steady state of operation is achieved when a known set of data repositories are being incrementally spidered and no major changes are being made to the volume of data being processed each cycle.

To acquire a solid understanding of how your server infrastructure is operating, it is strongly recommended you implement performance logging of your Lotus Discovery Servers soon after they are deployed.

When planning how to integrate backup of your Lotus Discovery Servers into your company's backup policy, there are some features of a Lotus Discovery Server which need to be taken into account. In addition, there are issues which relate to how you have configured your Lotus Discovery Servers and what load they are subjected to. For example, if you have decided to schedule execution of resource-intensive services for overnight processing, you might find the available window for backing up your primary Lotus Discovery Servers (i.e. while the server is idle) is very limited.

Features to consider about your Lotus Discovery Servers are:

- Many of the Discovery Server services are very "resource-hungry" and can take a long time to complete processing of large newly spidered data repositories.
8. Planning a backup strategy for KDS

- The Lotus Discovery Server and DB2 services will need to be shut down if your backup software is unable to backup open files (or at least gracefully apply temporary locks on objects to be backed up). This downtime will reduce the time available for scheduling of overnight server tasks or, in the worst case, overrun into prime time when users expect access to the system.

- Network traffic between the primary and secondary KDS servers can be extremely high. If you are considering a strategy which includes backing up files across the network, make sure you take into account your network bandwidth capacity.

**Note:** Keep in mind the difference between resource requirements during initial server setup, and ongoing resource usage after your system has reached steady state operation. Also consider the impact of spidering large new data repositories. First-time spidering may take longer than expected and can generate significant load on your Lotus Discovery Servers and network.

**Tip:** Install a second network card on a dedicated subnet to avoid network traffic contention between regular Lotus Discovery Servers communications and backup operations over the network.

Always make sure you can recover to a known state of operation by ensuring you have a complete and valid backup of your system before making any configuration changes (no matter how small) to your Lotus Discovery Server environment.

### 8.7 Backing up DB2 using the DB2 Control Center

In a distributed installation of the Lotus Knowledge Discovery System, you only need to back up the DB2 K-map on the primary Lotus Discovery Server. (This may change in a future release of KDS, so keep an eye on this requirement to ensure your backup strategy remains valid.) If you use the DB2 Control Center backup facility to back up DB2, remember to integrate this procedure into your overall system backup plan.

The following procedures for backing up DB2 can be found in the online documentation for KDS. They are repeated here (with additional screen shots) to help administrators who are unfamiliar with DB2. The example described in 8.7.1, “How to use DB2 backup” concentrates on a database entity called CONTMAP. The same procedure should also be applied to the entity called SATCTLDB to ensure you have backed up the complete contents of DB2.
Decisions on where you store (and recover) your data from are going to depend on how your hardware is configured and how you access tape devices attached to the system (i.e. local, or across the network). If backing up to tape, consider creating a fast staging disk drive for hosting the initial creation of the DB2 backup file. This will reduce the time required to back up DB2—particularly if access to your tape device is slow. It also allows you to get your primary server back on line sooner. Transfer of the DB2 backup file to tape can then be scheduled independently from the operation of the primary server.

8.7.1 How to use DB2 backup

The following procedure, also shown in Figure 8-3 on page 244, describes how to back up your DB2 data store to a staging disk. The primary KDS server is called SPECKLEDHEN.

To perform a DB2 backup:
1. From the menu bar, click Start -> Programs -> IBM DB2 -> Control Center.
2. In the Control Center window, click the plus sign (+) sign to expand Systems -> SPECKLEDHEN -> Instances -> DB2 -> Databases.
3. Find and click CONTMAP (CONTMAP) to highlight it.
4. Right-click CONTMAP (CONTMAP), then choose Backup -> Database.
5. In the Backup Database window, click the Backup tab and confirm that Directories or Tapes is the selected Media type option.
6. Click the Browse button and then, in the Path Browser dialog that displays, select\enter a subdirectory path for the Backup.
Chapter 8. Planning a backup strategy for KDS

**Note:** Choose a folder outside of the \Lotus, \DB2 and \DB2CTLSV folders (e.g. \Backup\DB2).

Note: Choose a folder outside of the \Lotus, \DB2 and \DB2CTLSV folders (e.g. \Backup\DB2).

- Select an entry from the list.

<table>
<thead>
<tr>
<th>Object type</th>
<th>Operation</th>
<th>Date</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database</td>
<td>Backup</td>
<td>04/12/2001</td>
<td>10:40:10 AM</td>
</tr>
<tr>
<td>Database</td>
<td>Backup</td>
<td>04/12/2001</td>
<td>11:11:37 AM</td>
</tr>
</tbody>
</table>

Write down the date and time of the backup since you'll need to know these when the time comes to restore the backup. In this example, a backup was performed at 11:11:37 AM on 04/12/2001 and the backup file uses the time as part of its file name.

![Figure 8-4  DB2 Backup - Recording the date and time](image)

7. On the Options tab, ensure the Offline option is selected. **Note:** Selecting Offline stops users from connecting to the database while the backup request is processing (see Figure 8-5).

![Figure 8-5  Selecting Offline to stop user access to DB2](image)

8. Return to the Backup tab and click the **Backup Now** button.
   (To see the full command being executed, as shown in Figure 8-6, click the **Show Command** button. You can also save this as a script.)
9. To check the progress and status of the backup, select **Tools -> Journal** from the Control Center menu bar, or click the **Journal** icon, as shown in Figure 8-7 on page 246.

![Figure 8-6 Using show command to see backup script](image)

10. The time taken to back up your DB2 data store will depend on its size, the disk I/O performance of your server, and the performance of your destination drive and/or tape unit.

**Note:** It’s useful to collect statistics on the elapsed time and volume of data being backed up for each run. Use this information to accurately schedule downtime required to back up the system and anticipate when upgrades to disk and tape will be required.
8.7.2 How to use DB2 restore

This procedure is documented in the online documentation files that are installed during the installation of a Lotus Discovery Server. The procedure is very similar to that documented in 8.7.1, “How to use DB2 backup” on page 244. By now it is assumed you are familiar enough with the DB2 Control Center to run a DB2 restore.

8.8 Vendor solutions

There are a number of third-party software companies that offer backup software for DB2 and Domino (KDS uses the Domino engine as part of its underlying system architecture).

Table 8-3 lists some of the companies that offer, or plan to offer, backup software for Domino and/or DB2. Web addresses are provided for contacting the vendors.

<table>
<thead>
<tr>
<th>Vendor</th>
<th>Domino</th>
<th>DB2</th>
<th>Web address</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tivoli Storage Manager</td>
<td>Yes</td>
<td>Yes</td>
<td><a href="http://www.tivoli.com/tsm/">www.tivoli.com/tsm/</a></td>
</tr>
<tr>
<td>CommVault</td>
<td></td>
<td></td>
<td><a href="http://www.comvault.com">www.comvault.com</a></td>
</tr>
<tr>
<td>ArcServeIT</td>
<td></td>
<td></td>
<td><a href="http://www.cai.com/arcserveit/">www.cai.com/arcserveit/</a></td>
</tr>
<tr>
<td>Legato</td>
<td></td>
<td></td>
<td><a href="http://www.legato.com">www.legato.com</a></td>
</tr>
<tr>
<td>Arkeia Backup Agent</td>
<td></td>
<td></td>
<td><a href="http://www.arkeia.com">www.arkeia.com</a></td>
</tr>
</tbody>
</table>

8.9 Features to look for in backup software

As part of the process of selecting backup software for backing up your Lotus Discovery Servers, you should consider the following issues:

- A distributed Lotus Discovery Server implementation can generate significant traffic between servers—especially during the initial setup, configuration, and spider of new data repositories. Will you be backing up across the network, and if so, do you have sufficient bandwidth to handle the load?
- First-time spidering of data repositories can run for a long time (several days, in extreme cases). Will the backup software yield if it detects active processes are still running on the server? (You won’t want to lose work processed so far by shutting down or stopping the Lotus Discovery Server.)
- Will the backup software handle problems like file locking, open files, timeouts, and so on?
Running backups concurrently with Discovery Services will significantly degrade server performance.

The primary Lotus Discovery Server server can quickly consume large amounts of disk space, so how much data do you expect to back up, and how long will it take?

What programming interfaces are available with the backup software?

Can the backup software be programmed to allow for special shutdown and restart requirements of a distributed Lotus Discovery Server implementation? (For example, the primary server is shut down last and restarted first.)

Backup media costs can be substantial if you deploy a dedicated tape unit per Lotus Discovery Server.

How easy is it to install, configure, and operate the backup software?

How much training and rewriting of backup procedures will be required?

What is the throughput capacity of the software, and do you have the necessary infrastructure to leverage this?

8.10 Running a selective backup

To perform a selective backup of all DB2 and Discovery Server files on the primary server, select the contents of the DB2, DB2CTLSV, and Lotus folders. For an installation where the Discovery services are distributed across several secondary servers, you must understand what data files are associated with each Discovery Service. When you have figured this out, you can then select the correct folder(s) and files to back up on each Lotus Discovery Server. For example, to back up a secondary server which hosts the K-map Indexing Service, you need to select all the following:

- The FTDOMAIN.DI folder, plus all sub-folders
- Indexmap.nsf (a Notes database under the Lotus\KDS\data folder)
- LDSFullTextRestart.00 through .07 (under the Lotus\KDS\data folder)
File types you can spider

This appendix includes a table with the file types supported by the file system spider in V1.0.

KeyView file viewer specifications

<table>
<thead>
<tr>
<th>Document format</th>
<th>Version(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASCII Text</td>
<td>All</td>
</tr>
<tr>
<td>ANSI Text</td>
<td>All</td>
</tr>
<tr>
<td>Applix Words</td>
<td>v4.2, 4.3, 4.4</td>
</tr>
<tr>
<td>HTML</td>
<td>v1.x, 2.x, 3.x</td>
</tr>
<tr>
<td>IBM DCA/RFT</td>
<td>vSC23-0758-1</td>
</tr>
<tr>
<td>IBM Displaywrite</td>
<td>v4, 5</td>
</tr>
<tr>
<td>Lotus AMI Pro</td>
<td>v2, 3</td>
</tr>
<tr>
<td>Lotus AMI Professional Write Plus</td>
<td></td>
</tr>
<tr>
<td>Lotus Word Pro</td>
<td>v96, 97, R9</td>
</tr>
<tr>
<td>Document format</td>
<td>Version(s)</td>
</tr>
<tr>
<td>-------------------------------------</td>
<td>--------------</td>
</tr>
<tr>
<td>Maker Interchange Format (MIF)</td>
<td>v5.5</td>
</tr>
<tr>
<td>Microsoft RTF</td>
<td></td>
</tr>
<tr>
<td>Microsoft Word for PC</td>
<td>v2.to 5.5</td>
</tr>
<tr>
<td>Microsoft Word for Macintosh</td>
<td>v4, 5, 6, 98</td>
</tr>
<tr>
<td>Microsoft Word for Windows</td>
<td>v2.x, 6.0</td>
</tr>
<tr>
<td>Microsoft Word for Windows</td>
<td>95, 97, 2000</td>
</tr>
<tr>
<td>Microsoft Works</td>
<td>v1.0, 2.0, 3.0, 4.0</td>
</tr>
<tr>
<td>Microsoft Windows Write</td>
<td>v1.0, 2.0, 3.0</td>
</tr>
<tr>
<td>Unicode Text</td>
<td></td>
</tr>
<tr>
<td>WordPerfect</td>
<td>v5.x, 6, 7, 8</td>
</tr>
<tr>
<td>WordPerfect for Macintosh</td>
<td>v2, 3</td>
</tr>
<tr>
<td>XyWrite</td>
<td>v4.12</td>
</tr>
<tr>
<td>Lotus Word Pro*</td>
<td>V96, 97, R9</td>
</tr>
<tr>
<td>Lotus 1-2-3*</td>
<td>V96, 97, R9</td>
</tr>
<tr>
<td>Lotus Freelance *</td>
<td>V96, 97, R9</td>
</tr>
<tr>
<td>Microsoft Word **</td>
<td>V6, 95, 97, 2000</td>
</tr>
<tr>
<td>Microsoft Excel **</td>
<td>V6, 95, 97, 2000</td>
</tr>
<tr>
<td>Microsoft PowerPoint **</td>
<td>V6, 95, 97, 2000</td>
</tr>
<tr>
<td>Applix Spreadsheets</td>
<td>v4.3, 4.4</td>
</tr>
<tr>
<td>Corel QuattroPro</td>
<td>v7, 8</td>
</tr>
<tr>
<td>Lotus 1-2-3</td>
<td>v2, 3, 4, 5, 96, 97, R9</td>
</tr>
<tr>
<td>Microsoft Excel</td>
<td>v3, 4, 5, 96, 97, 2000</td>
</tr>
<tr>
<td>Microsoft Excel for Mac</td>
<td>98</td>
</tr>
<tr>
<td>Microsoft Works Spreadsheet</td>
<td>v1.0, 2.0, 3.0, 4.0</td>
</tr>
<tr>
<td>Applix Presents</td>
<td>v4.3, 4.4</td>
</tr>
<tr>
<td>Corel Presentations</td>
<td>v7.0, 8.0</td>
</tr>
<tr>
<td>Lotus Freelance</td>
<td>v96, 97, R9</td>
</tr>
<tr>
<td>Document format</td>
<td>Version(s)</td>
</tr>
<tr>
<td>-----------------------------------------------------</td>
<td>--------------</td>
</tr>
<tr>
<td>Microsoft PowerPoint</td>
<td>v4.0, 95, 97, 2000</td>
</tr>
<tr>
<td>Microsoft PowerPoint for Macintosh</td>
<td>98</td>
</tr>
<tr>
<td>Graphics formats (not used for XML conversion in LDS)</td>
<td></td>
</tr>
<tr>
<td>AMI Draw Graphics (SDW)</td>
<td></td>
</tr>
<tr>
<td>Applix Graphics</td>
<td>v4.3, 4.4</td>
</tr>
<tr>
<td>Fax Systems (CCITT) Groups</td>
<td>3 &amp; 4</td>
</tr>
<tr>
<td>Computer Graphics Metafile (CGM)</td>
<td></td>
</tr>
<tr>
<td>Corel Draw CDR (TIFF Header)</td>
<td></td>
</tr>
<tr>
<td>Encapsulated PostScript (EPS)</td>
<td></td>
</tr>
<tr>
<td>Enhanced Metafile (EMF)</td>
<td></td>
</tr>
<tr>
<td>JPEG File Interchange Format</td>
<td></td>
</tr>
<tr>
<td>Lotus Pic (PIC)</td>
<td></td>
</tr>
<tr>
<td>Mac PICT (raster content)</td>
<td></td>
</tr>
<tr>
<td>MacPaint (MAC)</td>
<td></td>
</tr>
<tr>
<td>Microsoft Excel Charts</td>
<td></td>
</tr>
<tr>
<td>Microsoft Windows Animated Cursor</td>
<td></td>
</tr>
<tr>
<td>Microsoft Windows Bitmap (BMP)</td>
<td></td>
</tr>
<tr>
<td>Microsoft Windows Cursor/Icon</td>
<td></td>
</tr>
<tr>
<td>Microsoft Windows Metafile (WMF)</td>
<td></td>
</tr>
<tr>
<td>PC PaintBrush (PCX)</td>
<td></td>
</tr>
<tr>
<td>Portable Network Graphics (PNG)</td>
<td></td>
</tr>
<tr>
<td>Sun Raster</td>
<td></td>
</tr>
<tr>
<td>SGI RGB</td>
<td></td>
</tr>
<tr>
<td>Truevision Targa</td>
<td></td>
</tr>
<tr>
<td>WordPerfect Graphics (WPG)</td>
<td>v1, 2</td>
</tr>
<tr>
<td>Multimedia formats (not used for XML conversion in LDS)</td>
<td></td>
</tr>
<tr>
<td>Audio Interchange File Format (AIFF)</td>
<td></td>
</tr>
<tr>
<td>Document format</td>
<td>Version(s)</td>
</tr>
<tr>
<td>---------------------------------</td>
<td>------------</td>
</tr>
<tr>
<td>Microsoft Sound (WAV)</td>
<td></td>
</tr>
<tr>
<td>MIDI (MID)</td>
<td></td>
</tr>
<tr>
<td>MPEG 1 Video (MPG)</td>
<td></td>
</tr>
<tr>
<td>NeXT/Sun Audio (AU)</td>
<td></td>
</tr>
<tr>
<td>QuickTime Movie (MOV)</td>
<td></td>
</tr>
<tr>
<td>Video for Windows (AVI)</td>
<td></td>
</tr>
<tr>
<td>Fax Formats</td>
<td></td>
</tr>
<tr>
<td>DCX</td>
<td></td>
</tr>
</tbody>
</table>

**Compression, encapsulation and security formats**

<table>
<thead>
<tr>
<th>Compression, encapsulation and security formats</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>GZ-compression</td>
<td></td>
</tr>
<tr>
<td>Z-compression</td>
<td></td>
</tr>
<tr>
<td>ZipFax Systems (CCITT)</td>
<td></td>
</tr>
<tr>
<td>BinHex</td>
<td></td>
</tr>
<tr>
<td>MIME</td>
<td></td>
</tr>
<tr>
<td>TAR</td>
<td></td>
</tr>
<tr>
<td>UUencode</td>
<td></td>
</tr>
</tbody>
</table>

*Japanese

**Japanese, Korean, Simplified and Traditional Chinese
Notes.ini settings

This appendix provides the necessary notes.ini settings for Lotus Discovery Server V1.0.

### Notes.ini settings

**Table B-1  Notes.ini Settings**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>QueueCryptoLevel</td>
<td>Enables the spider queue encryption setting as follows:</td>
</tr>
<tr>
<td></td>
<td>To enable spider work queue encryption, enter</td>
</tr>
<tr>
<td></td>
<td>QueueCryptoLevel= high.</td>
</tr>
<tr>
<td></td>
<td>To disable queue encryption, enter</td>
</tr>
<tr>
<td></td>
<td>QueueCryptoLevel= none or leave it blank.</td>
</tr>
<tr>
<td>Note:</td>
<td>The setting is not case sensitive.</td>
</tr>
</tbody>
</table>

© Copyright IBM Corp. 2001
Inside the Lotus Discovery Server

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ExternalDominoDomain</td>
<td>List a server in an external Domino domain as a Notes.ini variable (one line for each external domain):</td>
</tr>
<tr>
<td></td>
<td>ExternalDominoDomain1=&lt;Domain name&gt;,&lt;Servername&gt;, such as:</td>
</tr>
<tr>
<td></td>
<td>ExternalDominoDomain=Acme, ServerA/Acme</td>
</tr>
<tr>
<td></td>
<td>ExternalDominoDomain2= etc.</td>
</tr>
<tr>
<td></td>
<td>At user-authentication time, KDS accesses this server for the information and creates the group lists.</td>
</tr>
<tr>
<td>KDS_CURSOR_MEMORY</td>
<td>The maximum amount of data to cache in memory in megabytes. The default is 10MB.</td>
</tr>
<tr>
<td>KDS_CURSOR_DISK</td>
<td>The maximum amount of data to spill to disk in megabytes. The default is 100MB.</td>
</tr>
<tr>
<td>KDS_CURSOR_TIME</td>
<td>The amount of time in minutes to hold data in the cache. The default is 60 minutes. This setting affects how quickly new data, people, and places show up in the K-map documents, people, and places lists.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> Leaving the cache time high can improve performance for users.</td>
</tr>
<tr>
<td>KDS_IOR_HOST</td>
<td>The IP address of the host used when building the IOR. This setting handles the case in which a system running KDS has two network cards, ensuring that the IP address for the correct network adapter is used.</td>
</tr>
<tr>
<td>NTMasterListRefreshInterval</td>
<td>A table of groups in the Windows NT/2000 domain listing the members in each group. This setting determines the amount of time in minutes between refreshes of the table’s content. This table needs to be rebuilt periodically to take into account any changes in the NT domain groups.</td>
</tr>
<tr>
<td>Variable</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>QPNameListRefreshInterval</td>
<td>A table of QuickPlace names in the primary server's Domino domain. This table needs to be rebuilt periodically to take into account any QuickPlace additions or deletions. <strong>Note:</strong> KDS doesn't support QuickPlaces in Domino domains external to the primary KDS server's Domino domain.</td>
</tr>
<tr>
<td>access_cache_life_minutes=60</td>
<td>All ACLs (security information about documents and databases) are cached. Information in the cache expires either when:  ► One hour has passed since it was put into the cache (not since last access).  ► The total amount of data in the cache is greater than 110 MB, expiring the least recently used data until it gets under the limit.</td>
</tr>
<tr>
<td>YSCHED_PULSE_INTERVAL= &lt;number of 10 millisecond ticks&gt;</td>
<td>This specifies the permissible length of time between spider messages before the Scheduler will mark the repository as overdue.</td>
</tr>
<tr>
<td>ProxyUserId</td>
<td>The settings where the user account that controls access to the proxy server is maintained.</td>
</tr>
<tr>
<td>ProxyUserPswd</td>
<td>The settings where the user account that controls access to the proxy server is maintained.</td>
</tr>
<tr>
<td>kstationbackup</td>
<td>Used by an upgrade install when upgrading from a K-station 1.0 installation to a full KDS installation with Discovery Server and K-station, to a partial KDS installation with just K-station, or to a K-station 1.0a installation. If KDS Install detects the previous version of K-station, it does the following:  ► Performs a backup of data files.  ► Creates the registry key &quot;HKEY_LOCAL_MACHINE\Software\Lotus\KstationBackup&quot; which stores the location of the backup directory.  ► Creates this kstationbackup variable and sets it to the path of the backup directory obtained from the registry</td>
</tr>
<tr>
<td>Variable</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>LPSSO=/lp/lpmaint.nsf/fa_login?Open Form</td>
<td>Enable Single Sign-on by default on the KDS server</td>
</tr>
<tr>
<td>FT_INDEX_ACCENT_SENS=1</td>
<td>Support search for accented characters (ö, ü, ø, å, ñ etc.)</td>
</tr>
<tr>
<td>KDS_WEBSPIDER_WARN_LEVEL</td>
<td>Change the detail level of the webspider log file:</td>
</tr>
<tr>
<td></td>
<td>1=log nothing</td>
</tr>
<tr>
<td></td>
<td>2=filter the logged messages</td>
</tr>
<tr>
<td></td>
<td>3=log everything</td>
</tr>
</tbody>
</table>
Additional material

This 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 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/SG246252

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, SG246252.
Using the Web material

The additional Web material that accompanies this redbook includes the following files:

<table>
<thead>
<tr>
<th>File name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>kdstools.zip</td>
<td>Contains a Notes database with a simple agent to change</td>
</tr>
<tr>
<td></td>
<td>author fields on documents to assist with testing affinities.</td>
</tr>
</tbody>
</table>

System requirements for downloading the Web material

The following system configuration is recommended:

- **Hard disk space:** 360 Kb
- **Operating System:** Any supported Notes platform
- **Processor:** Any supported by Notes
- **Memory:** Any supported by Notes

How to use the Web material

Create a subdirectory (folder) on your workstation under the \Data directory, 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 redbook.

IBM Redbooks

For information on ordering these publications, see “How to get IBM Redbooks” on page 261.

- Java 2 Network Security, SG24-2109
- TCP/IP Tutorial and Technical Overview, GG24-3376
- Customizing Quickplace, SG24-6000
- Lotus Sametime 2.0 Deployment Guide, SG24-6206
- Inside the Lotus Discovery Server, SG24-6252
- iNotes Web Access, SG24-6518
- Using Domino Workflow, SG24-5963
- XML Powered by Domino, SG24-6207
- Lotus Sametime Application Development Guide, SG24-5651
- COM Together — with Domino, SG24-5670
- Lotus Notes and Domino Take Center Stage, SG24-5630
- Performance Considerations for Domino Applications, SG24-5602
- Lotus Domino Release 5.0: A Developer's Handbook, SG24-5331
- Connecting Domino to the Enterprise Using Java, SG24-5425
- LotusScript for Visual Basic Programmers, SG24-4856
- Developing Web Applications Using Lotus Notes Designer for Domino 4.6, SG24-2183
- Lotus Notes 4.5: A Developers Handbook, SG24-4876
- Lotus Solutions for the Enterprise, Volume 1. Lotus Notes: An Enterprise Application Platform, SG24-4837
- Lotus Solutions for the Enterprise, Volume 2. Using DB2 in a Domino Environment, SG24-4918
Other Lotus-related ITSO publications

The publications listed in this section may also be of interest:

- **Lotus Solutions for the Enterprise, Volume 4. Lotus Notes and the MQSeries Enterprise Integrator**, SG24-2217
- **Lotus Solutions for the Enterprise, Volume 5. NotesPump, the Enterprise Data Mover**, SG24-5255
- **Enterprise Integration with Domino for S/390**, SG24-5150

- **A Roadmap for Deploying Domino in the Organization**, SG24-5617
- **The Three Steps to Super.Human.Software: Compare, Coexist, Migrate; From Microsoft Exchange to Lotus Domino, Part Two: Coexistence and Migration**, SG24-5615
- **Lotus Notes and Domino R5.0 Security Infrastructure Revealed**, SG24-5341
- **Lotus Notes and Domino: The Next Generation in Messaging. Moving from Microsoft Mail to Lotus Notes and Domino**, SG24-5152
- **Eight Steps to a Successful Messaging Migration: A Planning Guide for Migrating to Lotus Notes and Domino**, SG24-5335
- **Deploying Domino in an S/390 Environment**, SG24-2182
- **The Next Step in Messaging: Upgrade Case Studies on Lotus cc:Mail to Lotus Domino and Lotus Notes**, SG24-5100
- **Lotus Notes and Domino: The Next Generation in Messaging. Moving from Novell GroupWise to Lotus Notes and Domino**, SG24-5321
- **High Availability and Scalability with Domino Clustering and Partitioning on Windows NT**, SG24-5141
- **From Client/Server to Network Computing, A Migration to Domino**, SG24-5087
- **Netfinity and Domino R5.0 Integration Guide**, SG24-5313
- **Lotus Domino R5 for IBM RS/6000**, SG24-5138
- **Lotus Domino Release 4.6 on IBM RS/6000: Installation, Customization and Administration**, SG24-4694
- **High Availability and Scalability with Domino Clustering and Partitioning on AIX**, SG24-5163
- *Lotus Domino for AS/400: Installation, Customization and Administration*, SG24-5181
- *Lotus Domino for S/390 Performance Tuning and Capacity Planning*, SG24-5149
- *Porting C Applications to Lotus Domino on S/390*, SG24-2092
- *Measuring Lotus Notes Response Times with Tivoli’s ARM Agents*, SG24-4787
- *Using Tivoli Storage Manager to Back Up Lotus Notes*, SG24-4534

**How to get IBM Redbooks**

Search for additional Redbooks or redpieces, view, download, or order hardcopy from the Redbooks Web site:

[ibm.com/redbooks](http://ibm.com/redbooks)

Also download additional materials (code samples or diskette/CD-ROM images) from this Redbooks site.

Redpieces are Redbooks in progress; not all Redbooks become redpieces and sometimes just a few chapters will be published this way. The intent is to get the information out much quicker than the formal publishing process allows.

**IBM Redbooks collections**

Redbooks are also available on CD-ROMs. Click the CD-ROMs button on the Redbooks Web site for information about all the CD-ROMs offered, as well as updates and formats.
Special notices

References in this publication to IBM products, programs or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only IBM's product, program, or service may be used. Any functionally equivalent program that does not infringe any of IBM's intellectual property rights may be used instead of the IBM product, program or service.

Information in this book was developed in conjunction with use of the equipment specified, and is limited in application to those specific hardware and software products and levels.

IBM may have patents or pending patent applications covering subject matter 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 the IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact IBM Corporation, Dept. 600A, Mail Drop 1329, Somers, NY 10589 USA.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The information contained in this document has not been submitted to any formal IBM test and is distributed AS IS. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer's ability to evaluate and integrate them into the customer's operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environments do so at their own risk.

Any pointers in this publication to external Web sites are provided for convenience only and do not in any manner serve as an endorsement of these Web sites.
The following terms are trademarks of other companies:

Tivoli, Manage. Anything. Anywhere., The Power To Manage., Anything. Anywhere., TME, NetView, Cross-Site, Tivoli Ready, Tivoli Certified, Planet Tivoli, and Tivoli Enterprise are trademarks or registered trademarks of Tivoli Systems Inc., an IBM company, in the United States, other countries, or both. In Denmark, Tivoli is a trademark licensed from Kjøbenhavns Sommer - Tivoli A/S.

C-bus is a trademark of Corollary, Inc. in the United States and/or other countries.

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

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States and/or other countries.

PC Direct is a trademark of Ziff Communications Company in the United States and/or other countries and is used by IBM Corporation under license.

ActionMedia, LANDesk, MMX, Pentium and ProShare are trademarks of Intel Corporation in the United States and/or other countries.

UNIX is a registered trademark in the United States and other countries licensed exclusively through The Open Group.

SET, SET Secure Electronic Transaction, and the SET Logo are trademarks owned by SET Secure Electronic Transaction LLC.

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

Symbols
$Global  64

A
Active Directory
  user profiles  5
Affinities  68
  description  4
  problems with  225
  user control of  8
  what are they?  7
Affinity
  from metrics  8
  viewing with the K-map editor  9
AffinityCalcTask  51
AffinityWorkerTask  51
Amgr  77
ATG  70, 78
  K-map editor  58
ATGOutOfBandClassifyQ. Queue  48
Authentication  165
Authorization  165

B
Backup  235
  distributed servers  239
  open files  240
  strategy  236
  to tape  237
BM SABIO Tools  57
Business case
  preparing  19

C
Categorizer  69
Chief Knowledge Officer  21
CMap
  see content map  59
Collection
  metrics  51
Completion Queue  42
CompletionQ. Queue  48
Consistent ACL  159
Content map  59
Cookies  168
CORBA  164
CPU performance  188
Crashes  209

D
dashboard.nsf  39, 217
Data custodians  17
Data repository  4
Data sources
  selecting  31
Database ACL security  158
DB2  2, 182
  backing up  243
db2syscs.exe  212
Deployment
  getting started  14
  planning  13
Diagnosing problems  211
DIIOP  39, 78
DirectorySync  50
DirectorySyncWorkQ  49
Discovery Server
  components  37
  shutdown  39
  tasks  38
Discovery Server Control Center
  description  3
Discovery Services  38
DNS  223
Document fit value  52
Document scoring  51
Document Security  162
Document valuation
  description  5
Domain Search  9
Domino security  153
Domino.Doc  4, 41
Inside the Lotus Discovery Server

**E**
Encryption
documents 162
End user 26

**F**
Field maps 64
Field security 163
File system spider 4, 46
supported file types 46
FileSyncSpider 79
FileSyncSpiderQ.Queue 49
FT_INDEX_ACCENT_SENS 71
FullText
  K-map editor 58
FulltextQueue 42

**H**
Hardware 35, 187
requirements 10, 76
HTTP 78

**I**
I/O performance 186
I/O Reads 212
InXight 2, 56, 61
InXightTools 70

**K**
KDS_BOA_THREADS 202
KDS_CURSOR_DISK 60
KDS_CURSOR_MEMORY 60
KDS_CURSOR_TIME 60
KDS_EVENT_THREADS 202
KDSAdministrators 229
KeyView 2
K-map 2, 53
  access to 9
  backing up 241
  building 54
  building recommendations 63
  considerations before building 33
  description 3
  document clustering 8
  editing 58
  editor 3
  IBM SABIO tools 57
  indexing 9, 66
  maintaining 18
  problems 227
  searching 59
  stopword list 53
  viewing affinities 9
K-map Builder 3
K-map building 78, 183
K-map Editor 2, 58
K-map indexing 78, 183
Knowledge Audit 15
Knowledge Discovery System 13
Knowledge Management 2
  strategy 14
Knowledge map
  see K-map 3
K-station 2, 184

**L**
LanguageWare 71
LDAP 48, 78, 165
user profiles 5
LDS
  see Lotus Discovery System 2
LDSATGCmapWorkQ.Queue 48
LDSFullText 203
LDSFullTextWorkQ.Queue 49
LDSFullTextWorkQ.queue 66
LDSMetrics 203
LDSMetricsWorkQ.Queue 48
LDSProfileSyncWorkQ.Queue 49
LDSTaxonomy 203
LDSTaxonomyWorkQ.Queue 49
LDSTaxonomyWorkQ.queue 54
Legacy data 28
Log files 225
Lotus Discovery Server
  system requirements 10
  what is it? 7
Lotus Knowledge Discovery System 2

**M**
Mail files 17
Memory 188
Meta data
  determining which data to use 16
Metrics 50, 68, 183
  collection 8, 51
document valuation 5
K-map editor 58
processing 50
what are they? 7
Metrics Collection 79
Metrics Collector 3
Metrics evaluator 23
Metrics Processing 79
Metrics Processor 3
MetricsDirectorySyncWorker 49
MetricsQueue 42
MetricsWorker 51
MetricsWorkQ 51
Monitoring
  CPU 196
disk performance 198
Monitoring performance 196
Multi-language 70

N
natg.exe 50, 212
ncmserv.exe 212
ndssched.exe 50
Network cards 188
nfilesysspider.exe 50
nkmapbuild.exe 50
nkmapindex.exe 50
nmetrics.exe 50
nnotesspider.exe 50
Notes spider 4, 47
Notes.ini 253
  FT_INDEX_ACCENT_SENS 71
  KDS_BOA_THREADS 202
  KDS_CURSOR_DISK 60
  KDS_CURSOR_MEMORY 60
  KDS_CURSOR_TIME 60
  KDS_EVENT_THREADS 202
  performance settings 202
NotesSpider 79
NotesSpiderQ.Queue 49
nportalsetup.exe 50
nwebspider.exe 50

O
ORB 78

P
Paragraph security 162
People.nsf 77
people.nsf 49
Performance 181
  startup versus ongoing requirements 11
Pilot
  preparing for 20
Planning
  a Discovery Server system 182
    for deployment 13
Primary KDS server 183
Privacy 51
  policy 18
Problems
  top 10 221
Profile Source 3
Profile source 78
Profile source spider 47
Profile synchronization 67, 78
Profile Synchronizer 3
ProfileSource spider
  metadata 48
ProfileSpiderQ.Queue 49
Public Access users 162

Q
Queue
  handling 48
QuickPlace 41
  problems 228
Quickplace 77

R
Recovery 205
  plan 206
Redbooks Web site 261
  Contact us xi
Replica 78
Router 78

S
Sametime 2, 77
Secondary KDS servers 183
Secureway Directory
  user profiles 5
Security 33, 149
Inside the Lotus Discovery Server
Inside the Lotus Discovery Server

Understanding the KDS architecture

The Lotus Discovery Server is a knowledge server. It provides search and expertise location solutions designed to ensure that the relevant knowledge and collective experience of an organization is readily available to help individuals and teams solve everyday business problems.

In this IBM Redbook, we describe how to set up a successful Discovery Server implementation in an organization. We show you how to begin the project by looking at and sorting through the information you already have, how to install a pilot system you can use to learn how Discovery Server works with your information, and how to design your system to work effectively across multiple servers. We also touch on how this complex product works under the covers.

The Discovery Server extracts, analyzes and categorizes structured and unstructured information, revealing the relationships between content, people, topics and user activity in an organization. It automatically generates and maintains a Knowledge map (K-map) to display relevant content categories and their appropriate hierarchical mapping, which can easily be searched or browsed by users. It also generates and maintains user profiles and tracks relevant end-user activities, identifying individuals who may be subject matter experts. The Discovery Server uncovers organizational know-how.

Setting up a pilot

Deployment tips

For more information:
ibm.com/redbooks