Collaborative document management for the enterprise with IBM Lotus Domino Document Manager

By Luke Raiano, Product Manager, and Kioko Mwosa, WW Technical Sales
Lotus Software
IBM Software Group

August 2005
# Table of Contents

3  Executive summary  
4  Part I – Capabilities and Future Direction  
12  Part II – Deploying Lotus Domino Document Manager  
30  Lessons Learned  
31  References
Executive Summary
For many businesses, the primary information management tool that users have at their disposal is the email in-box. Documents come in and go out with little thought as to how best to leverage the information that they contain or to what controls might be appropriate to provide security and management capability. But it’s this critical information that’s driving modern businesses. And it’s not only the documents in e-mail in-boxes, but also those scattered on laptops, on file servers and in paper file cabinets, making them difficult to find and re-use. That means time is wasted reproducing information that already exists. The result? Lost intellectual capital, lost productivity, increased storage costs and missed opportunities.

Today’s organizations require flexibility and cost effectiveness in the document management system that they use. In other words, they need to:

- Provide an easy to use set of tools to enable users to collaborate effectively with minimal training
- Improve the operational efficiency and productivity of the users and organization that it serves
- Reduce administration and production costs by taking advantage of technology investments already made
- Provide the scalability, reliability and manageability needed for cost-effective, enterprise-wide deployment
- Automate support for distributed management of diverse content throughout its life cycle, from authoring through review, approval, distribution and archiving.
- Work with a trusted vendor
- Utilize an open, extensible architecture enabling organizations to quickly extend document management capabilities to new business needs and broader strategic initiatives
- Nurture a broad spectrum of partner technologies to help accelerate the return on investment

Built on the ideal foundation of IBM Lotus Domino messaging and collaborative architecture, IBM Lotus Domino Document Manager is unique in its ability to help organizations meet all the above requirements for an effective document management system.

This white paper is divided into two parts. The first part presents an overview of the IBM Lotus Domino Document Manager architecture, and explores how it can best be used to leverage existing investments in IBM Lotus technologies and streamline organizational processes. The second part provides a best practices guide on how best to implement Lotus Domino Document Manager as well as some of the lessons learned in previous deployments.
“Managing” Information Today

For many of us, the primary information management tool that we have at our disposal is our email inbox. Documents come in and go out with little thought as to how best to leverage the information that they contain or to what controls might be appropriate to provide security and management capability. To understand the business value of document management, let’s consider a “day in the life” scenario of Tim, our Marketing Communications Specialist.

Just like you and me, Tim’s primary tool for managing information is email. Let’s say Tim is asked to put together a specification sheet for the company’s newest product. He generates the document using his word processor and emails it out to Sarah, John and Bob for review.

Each of them receives a copy and makes additions, deletions and comments and sends them back to Tim so that he can incorporate all of those changes back into one document. Once he’s made those changes, and perhaps some additional modifications, Tim then sends the specification sheet to the company’s sales force for consumption.

Meanwhile, Sarah has been asked to give a presentation to a group of the company’s key business partners. So, she utilizes the information in the specification sheet that she reviewed earlier for Tim with the changes that she made -- oblivious to the changes that John, Bob and Tim have made.

At the same time, Bob is going to a trade show and he’s going to use the information contained in his copy of the data sheet, not knowing about the changes that the others have made.

So we have five different versions of the same document floating around inside the company, and the information contained in each is making its way out to customers and business partners. This leads to many things, success not being one of them. The inconsistent information now floating inside and outside the company will take time and effort to sort out; time better spent doing other, more productive things.

This same scenario plays out in most organizations today. Ask yourself this question: Do you store most of the information that you use each day inside of your email file? I’ll bet your co-workers do. And, just think about the storage costs when 25 copies of the same presentation are saved all over the organization – in various versions -- when only one is needed. Have you ever gotten so frustrated at having the wrong information that you just gave up? I’ll bet your customers have too. All of these things affect the bottom line.

Integrated Messaging and Collaboration: The Ideal Starting Point

Lotus Domino already incorporates the underlying services that today’s businesses require for organization-wide document management as Figure 1 illustrates. Additionally, a single-vendor, integrated infrastructure for document management, messaging and
collaboration offers a key advantage over multi-component, multi-vendor options: attractive total cost of ownership and operation. Lotus Domino Document Manager offers enterprise-wide document management at a fraction of the cost of vertical document management solutions — often at a lower total cost than multi-vendor offerings — because critical components are built into Lotus Domino, like:

- A robust object store.
- The industry’s leading replication services, designed from the ground up to solve the problems of managing documents in a distributed environment.
- Proven, enterprise-strength security features, with everything from certificate-based authentication to SSL encryption to document-level access controls.
- An enterprise-scale, LDAP-accessible directory.
- A scalable, mail-enabled infrastructure.
- An integrated development environment.
- A full-text search engine capable of searching the entire object store.
- Support for standards-based protocols.

![Figure 1: The Lotus Document Management Infrastructure](image)

Thanks to the proven Domino infrastructure that supports it, Lotus Domino Document Manager fully addresses the fundamental requirements for enterprise-wide collaborative document management. And Lotus Domino Document Manager itself can be enhanced
via an array of complementary products from IBM and IBM Business Partners ranging from integrated imaging solutions to custom, value-added tools for specific customer requirements.

**Life Cycle Management**

Any document, whatever its structure or content, moves through a series of processes, which collectively define its “life cycle.” Generally speaking, all documents go through five distinct stages as part of their life cycle (some iteratively, some only once):

1. They are authored
2. They are reviewed and reworked
3. They are formally or informally approved
4. They are distributed or made available to a wider audience
5. Their utility drops below a preset level and they are archived or deleted

Figure 2 illustrates how the amount of activity (changes made, times opened for reading, etc.) associated with a document tends to diminish as it moves toward the end of its useful life. Eventually, a document no longer contributes to the day-to-day operation of the organization and can be archived for occasional access.

Life cycle management is relevant to all managed document content. It differs from related workflow processes in that life cycle events are triggered by events related to the status or properties of documents. Workflow events, in contrast, are triggered primarily by transactions; for example, a loan approval or an e-commerce purchase.

Life cycle management is integral to any enterprise-wide document management initiative, and represents a tremendous opportunity for:

- Greater control and consistency, by ensuring that all documents of a given type move through the same life cycle of processes.
- Cost savings through more efficient use of storage media.
- Increased efficiency, through automation of time-consuming routine processes.
Collaborative Authoring
The ease with which users can create documents with the desktop tools they already use makes Lotus Domino Document Manager a key productivity tool for collaborative authoring. When a user creates or stores a document in Lotus Domino Document Manager, it is assigned a document type. A document’s type determines how it is managed during its life cycle. For example, a document of type “contract” might need to be reviewed by representatives from Sales, Development and Legal; whereas a document of type “Standard Operating Procedure” might need to be reviewed by manufacturing and engineering.

To facilitate consistent handling of documents enterprise-wide, Lotus Domino Document Manager software stores document type information in the Library. File cabinets leverage this information by specifying what types of documents the file cabinet can store. This enables authors to utilize standard document types, guaranteeing consistent and predictable processes. Life cycle information is part of each document’s profile, so its status (“ready for review,” “in review,” “reviewed,” etc.) is easy to determine.

Once created, the file cabinet can be accessed by both contributors and consumers of document content. Creators can strictly control others’ ability to read, modify, add and/or delete documents both generally and individually within the file cabinet.

Review and Approval
Once a document has been authored, it often goes through an iterative process of review by peers and/or supervisors. Typically, input obtained during the review process results in a series of changes to the original document. Reviewers may modify the document themselves, or pass along comments, which authors then incorporate into the working document.

The revised document, incorporating changes made during the review process, then optionally moves to the Approval stage. Approval differs from review in that it involves no changes to the document’s content. The only action approvers perform is to either “approve” or “reject” the specific document. Lotus Domino Document Manager also allows approvers to include comments, to elaborate on their decision.

Integration with the messaging infrastructure, and with other collaborative functions, can help automate distribution of the document, routing to subsequent reviewers and approvers, and using the optional presence awareness integration with IBM Lotus Sametime®, initiate instant messaging sessions with relevant document stakeholders. Lotus Domino Document Manager provides several features to keep the review/approval process moving smoothly, including easy-to-specify expiration events and reminders.

Release
Once a final version has been approved, a document is released and generally available for users with at least reader access. Like the format, the properties of the released version of the document are also likely to differ from those of non-released versions or
drafts. In particular, security and access settings applicable during authoring and review may be too restrictive for the release version. The document management system must therefore be flexible enough to facilitate appropriate changes, such as storing released versions in a flatter, less tightly secured hierarchy.

To make it easier to control and manage renditions of a document (versions whose formats differ), Lotus Domino Document Manager allows documents reformatted for distribution to be stored alongside the original document. For example, when a document is viewed, a rendition (such as a PDF) may be displayed instead of the original. Lotus Domino Document Manager also automates important aspects of the release process. In particular, its built-in event triggers can automatically drive other external processes like format conversions, or external business processes.

**Archiving**

Once activity related to a document drops below a defined threshold, the document is ready to be moved out of the document management system into some alternate repository. Efficient archiving demands several important capabilities:

- A “proxy” or “stub” document specifying search attributes and other relevant information pertaining to the original must remain in the document management system for search and retrieval purposes.
- It must be easy to set archive parameters at the document level (i.e., via document types), so that all documents of a given type can be processed in a consistent manner.
- Interfaces to other processes must be available to automate storage to alternate repositories, to enable organizations to leverage current investments in hardware and applications.

Lotus Domino Document Manager provides rich functionality in all these areas. For example, it allows you to specify archival triggers that are user-initiated, time-driven or formula-based.

**Distributed Document Management**

The model for optimal functional capability for document life cycle management and many other functional areas changes profoundly in a distributed enterprise. A truly distributed and collaborative document management system must be inherently open and distributed across all of its basic functions:

- Management and securing of distributed content and associated workflows.
- Generation, capture, retrieval and/or publishing of document content from anywhere on the network.
- Storage and organization of distributed content.

Simply put, a distributed collaborative document management system allows content to be created collaboratively, with the user’s choice of tools. Once created, content must be
accessible via a variety of clients, from anywhere on the network. Ideally, content should be delivered from whatever location is closest to the requester.

The Lotus Domino Document Manager architecture ensures document integrity across a distributed network, thanks to its built-in document locking technology combined with the replication capabilities of Domino. These industrial-strength services enable Lotus Domino Document Manager to manage distributed content reliably and flexibly across the enterprise.

**Client Interfaces**
Distributed environments are by definition “virtual,” shifting whenever necessary across the boundaries of departments and organizations. Corporate or departmental “standards” are difficult to enforce in this context and open standards must therefore be adopted.

Since members of a virtual team may not be using the same client applications, the system must permit documents to be accessed and managed with a range of standards-based, familiar clients (as opposed to a single, proprietary client). Lotus Domino Document Manager supports access via:

- Web browsers
- Lotus Notes®
- Other POP3 mail clients
- Desktop applications (e.g., Microsoft® Office, Lotus SmartSuite®) via ODMA and other means (See Figure 3)
- Custom clients (via Lotus Domino Document Manager’s OLE automation API)
Email Integration

Many of us work within our email clients almost continually on a given day. This is generally how we receive the documents and content that we use to drive our businesses. A truly pervasive document management system will allow for integration into email so that users can seamlessly move content into the document store. Lotus Domino Document Manager provides for:

- Direct integration with Lotus Notes, without modification to the mail template
- Granular options to store content and attachments
- Optional trimming of attachments with a link to their location in the file cabinet

Integration with Other Applications

Users in a distributed environment may also need to interact with Lotus Domino Document Manager via the clients they use to access other applications; for example, custom clients for workflow, engineering design or customer support.

Lotus Domino Document Manager’s OLE Automation API is explicitly designed to allow customers and third parties to integrate Lotus Domino Document Manager with virtually any other application via Microsoft Visual Basic®, Visual C++™, LotusScript® and other programming tools that support OLE Automation. The API is also designed to support the creation of custom clients to access Lotus Domino Document Manager.
functionality.

The section “OLE Automation API” describes the Lotus Domino Document Manager API in greater detail.

Security
Robust security features are a well-established benefit of Domino, and hence of Lotus Domino Document Manager. A server-based challenge/response protocol is used to authenticate each user. Authentication is backed by a granular, administrator-defined authorization scheme:

- Access to each Lotus Domino Document Manager file cabinet is governed by an Access Control List (ACL), which specifies who can and cannot access the file cabinet. Users are specified by name or by membership in groups.
- Binder and document-level privileges further refine security, by restricting read and edit access to certain documents. For example, documents can be assigned a specific set of managers, authors and readers. Users not on any one of these lists will not be able to access the underlying content.
- Administrators and users can view a chronological audit trail of all the activities associated with a document such as the document creator, the creation date and time, and the sequence of authors who subsequently edited the document.

Integration via OLE Automation
Lotus Domino Document Manager supports integration with other applications via an OLE Automation API. The API provides access to Lotus Domino Document Manager from LotusScript, Visual Basic, Visual C++ and other programming tools or applications that support OLE Automation. It enables full programmatic access to the Lotus Domino Document Manager storage hierarchy and its functionality, such as creation of new documents in Lotus Domino Document Manager, document check-in/check-out, and modification of document security and profile information.

The API makes Lotus Domino Document Manager a document management service, which customers and third parties can use to:
- Integrate document management processes with custom applications like advanced workflow.
- Perform document management functions from non-ODMA desktop applications.
- Create custom, special-purpose clients or user agents.
- Extend the reach of Lotus Domino Document Manager’s Life Cycle Management capabilities.

Business Partner Offerings
IBM Business Partners offer a full complement of expertise and support to extend Lotus Domino Document Manager, from development of custom solutions, to integration of Lotus Domino Document Manager with current technologies, to off-the-shelf applications that add value to the Lotus Domino Document Manager infrastructure.
High-value solutions currently available from IBM Business Partners include integration with other document management applications, imaging, business-specific workflow applications, Web content creation, PDF and HTML rendering, and administration tools.


**Future Directions**

We believe that the value of a document management system is best enhanced by making it pervasive and tightly integrated with a user’s workspace experience. IBM envisions this convergence of collaboration and content within the framework of the IBM Workplace portfolio.

In this context, Lotus Domino Document Manager is expected to evolve to provide a number of innovations and technologies that will integrate it tightly with the IBM Workplace™ environment.

The Java Content Repository (JCR) is the implementation of Java Specification Request 170 (JSR 170), a standard for defining access to content. This enables the development of content based applications that can be deployed on top of a number of different underlying repositories. Plans call for Lotus Domino Document Manager to implement the JCR in several phases. Work is already well under way.

What this means for a customer as the IBM portfolio evolves is protection for their investment in Lotus Domino Document Manager. Put simply, documents stored in Lotus Domino Document Manager would be easily accessible to future generations of IBM software products.

Practically for the end user, this means seamless integration with IBM Workplace Client Technology™. As shown in Figure 3, a user using

**Ongoing Leadership**

For over 15 years, IBM has been delivering on an expanded vision for collaboration and improving the way people work. IBM continues to build on this vision today with the delivery of technology enhancements, new services and industry solutions that offer appropriate access to the people, processes and information to help customers transform their business to the IBM Workplace model.

**PART II – Deploying Lotus Domino Document Manager**

There is no one-size-fits-all strategy for deploying Lotus Domino Document Manager. IBM can offer many recommendations and suggestions, as detailed in this document. However, effective deployment of Lotus Domino Document Manager depends fundamentally on the skills of the administrators, system designers, and developers who can best configure each application/server combination to achieve optimum utility and performance. Deploying Lotus Domino Document Manager successfully also requires an
understanding of the business requirements and how they apply to a document management system.

The process of deploying Lotus Domino Document Manager typically consists of four phases:

1. Planning
2. Development
3. Staging
4. Production.

Each phase has a clearly defined objective. This section is intended to help you understand the implications of the choices you make in each phase. Much of the information included here comes from successful deployments of Lotus Domino Document Manager. Information was also gathered from questions and responses posted in many public discussion forums. Before taking action based on the information in this white paper, it is strongly recommended that you read and understand the Lotus Domino Document Manager Administrator’s Guide, and Lotus Domino Document Manager Programmer’s Guide. When installing or upgrading Lotus Domino Document Manager, follow the procedure listed in the Lotus Domino Document Manager Installation Guide. These resources contain additional useful information that is not included below.

Phase 1: Planning

The objective of the Planning phase is to clearly determine the architecture of the entire system.

General Areas to Consider
Giving careful thought to these questions below will you help identify and understand your organization’s document management requirements.

Geography
- How many sites will be participating?
- Which locations have the most users?
- Are there international sites?

Business units
- What are the major business units in the organization?
- Do the business units map to the geography and if so, how?
- Will users in one business unit need to access information in a different business unit’s repository, and if so, how frequently?

Projects
- What projects would benefit from using Lotus Domino Document Manager to manage their documents?
Notes domains
- Are there multiple Lotus Notes domains?
- Do the Lotus Notes domains map to the geography?

Corporate infrastructure
- How many documents need to be managed?
- How are documents currently created and stored now? Are they stored on a networked server?
- What is the existing hierarchy? Can it be mapped to the Lotus Domino Document Manager hierarchy?
- How are users going to access the system (for example, via the Web, a Lotus Notes client, or other means)?

Security
- What user roles need to be defined to add, delete, and update documents? What level of access does each role require?
- What documents, if any, contain secure information that needs to be restricted to particular users or groups? Should access to prior document revisions be restricted?
- Who should not have access to documents?
- Should security be managed centrally or can some autonomy be given to users outside of the IT department?

User groups
- What is the naming convention for the user groups?
- How are users added to and deleted from each group?
- When modifying users and groups, what kind of audit trails are necessary?
- How many users will have access to the system? How many users will be actively using the system on average? At peak usage? Will the users be primarily creating and editing documents or viewing them?

Application integration
- What applications can benefit from interacting with Lotus Domino Document Manager?
- Are these applications Open Document Management API (ODMA)-enabled?

Hardware Selection Guidelines
Capacity planning should be given careful consideration. The hardware on which Lotus Domino Document Manager is run should be sufficiently robust and scalable to allow upgrading as more data and users are added to the system.

CPUs
The server should have a minimum of two processors, but preferably more. Increasing the number of processors lets you take better advantage of the ability of Domino to distribute its resource load automatically across multiple processors. Multiple processors
can do more work in parallel.

**RAM**
The server should have a minimum of 1GB RAM. As with the number of processors, more is better.

**Disk storage**
You can use different storage systems to protect users’ data.

**RAID**
Redundant Array of Independent Drives (RAID) is a system that increases server performance and reliability by using multiple drives to store and manage data. There are different levels of RAID, i.e. level 0 - 5.

It is recommended that you use a RAID 1 disk array for Lotus Domino databases, and an expandable RAID 5 array for Lotus Domino Document Manager databases spread across multiple physical drives. To provide optimal performance, disk speed should be at least 10,000 rpm.

**SAN**
Storage Area Network (SAN) is an architecture for distributing data and can be used for functions such as clustering, shared repositories, and disaster recovery. Storing databases on a SAN lets you scale and access Lotus Domino Document Manager when working with up to several terabytes of data. To help manage system performance and scalability, it is recommended that you store databases containing larger files such as multimedia files and images on a SAN. For more information on designing a SAN architecture, read the IBM Redbook, “Designing an IBM Storage Area Network,” available online at http://www.redbooks.ibm.com/redbooks/SG245758.html.

**Network and bandwidth**
Any application that transfers large numbers of files will require high bandwidth for maximum performance. Where possible, use the fastest possible network cards for the server, network hardware scalable to 100 – 1000 Mbps, and use multiple LAN segments (one for each partition) to isolate network traffic at high end user loads. Over slower connections, it is recommended that you set up Lotus Domino Document Manager in a replicated environment with users accessing a replica library closest to where the work is being done; replication should be scheduled to occur when the fewest users are on the system.

**Availability**
A clustering and fail-over solution is available through IBM Business Partner, Carefree Technologies. More information about this solution can be found at: http://www.carefreetech.com.

When choosing your hardware, select a server that has additional failover on every hardware component (for example, multiple network cards, alternative network paths,
and so on).

**Backup and recovery**

It is good practice to schedule incremental backups on a regular basis. All system data should be backed up at least once a week, and high priority data should be backed up daily, or as appropriate for your environment.

The media used for file backup can vary from Optical Jukebox to less expensive media such as tape. Restoring data can be done on a per file basis or for an entire server using a backup system such as Tivoli® Storage Manager.

Another option is to create a replica library on a separate server and configure one-way replication. This ensures that data is copied to another location at a frequency which can be adjusted as needed. It also provides a level of redundancy by making documents available for read access.

**Operating System Considerations**

The Lotus Domino Document Manager server can run on multiple platforms. Check the documentation for the specific version you are deploying to determine what operating systems are supported. Each platform has benefits that may vary depending on existing IT policies and administrators’ skills sets.

**Using Windows Operating Systems**

Windows server operating systems are appropriate for small to medium-sized installations of Lotus Domino Document Manager. Typically, this platform works well in environments where the number of users is less than 2000, and the amount of data is less than 300GB. Running Lotus Domino Document Manager on the Windows platforms lets you use the Lotus Domino Document Manager API, an Object Linking and Embedding (OLE) Automation Server and its methods and properties. Only the Windows environment allows the Lotus Domino Document Manager API to run agents for processing data on the server.

A disadvantage to using Windows NT 4.0 specifically is the size of the page pool that is used to manage virtual memory. This cache is controlled by the paging file and can grow only to a maximum size of 192 MB. 1GB of data requires 1MB of page pool. If there is not enough RAM available for the operating system and data, more paging occurs, which subsequently slows down system performance and response time.

**Using UNIX®**

Both IBM AIX and Sun Solaris UNIX systems are appropriate for small to large installations of Lotus Domino Document Manager. UNIX may be the platform of choice for companies that are running several Lotus Domino applications in addition to Lotus Domino Document Manager on the same server.

A disadvantage to using the UNIX platform is that it may be difficult to integrate Lotus Domino Document Manager with other applications because the Lotus Domino
Document Manager API is not available on UNIX.

However, Lotus Domino Document Manager 6.5.1 integration with Lotus Workflow 6.5.1 is now supported on all major non-Windows platforms.

**Using OS/400**

IBM OS/400 is a scalable, stable, and high performing platform that is appropriate for medium to large installations of Lotus Domino Document Manager. The iSeries system scales well as you can have up to 12 processors running, use up to 40GB of main memory, and store up to 4TB worth of data. The cost of supporting this kind of system is low because of its stability. Up to 30 partitioned Domino servers can be supported on one system, which makes it easier to manage the servers. Lotus Domino Document Manager for iSeries provides specialized agents to archive and retrieve documents using Hierarchical Storage Management (HSM).

The main disadvantage to using OS/400 is that Lotus Domino Document Manager API is not available on OS/400.

**Phase 2: Development**

The main objective of the Development phase is to configure and customize Lotus Domino Document Manager according to the requirements identified in the preceding Planning phase. It is good practice to set up a development server on which developers can code and test the required document types, views, agents, forms, UI, and so on.

Another objective is to tune server performance. Performance measuring tools such as IBM Rational® Robot can be used to simulate user behavior and to load data to create performance benchmarks.

**Architecture**

Different companies have different uses for Lotus Domino Document Manager. Some companies use it as a back-end repository for documents that are accessed by other applications on the front-end; other companies use it as a replacement for their file system. Its primary purpose is to manage documents in one way or another through the authoring, review and approval, distribution, and consumption stages.

**Centralized vs. modular architecture**

Designing a centralized architecture model combines all the document management stages in a single structure, and allows all users to access the system from a single entry point. Policies can be standardized for the entire system while allowing for centralized backup and administration. There are some performance implications when using this model because all the system resources need to be shared for all user tasks and system tasks. For instance, someone who checks out multiple documents for editing uses the same system resources as someone else who searches for documents.
Defining a modular architecture (Figure 4) to correspond to each stage of document management is an effective way to design the system. This model uses separate physical servers for the life cycle: creation, editing, review and approval, distribution, and consumption stages. System users can be routed to a specific server to perform particular tasks. For example, if a user wants to create or edit content, that person can be routed to the appropriate server where he or she has editor access. If a user just wants to read content, that person can be routed to a different server where the content is published.

The benefit of using the modular architecture model is that it is easier to load balance and apply specific policies for each stage. Access and security policies can also be controlled more easily, depending on the particular server being used. However, a modular architecture can be costly as each stage requires its own hardware, backup, and management.

Figure 4: Modular architecture
Volumetric Analysis

Database size
The maximum size of each Lotus Domino Document Manager database should not exceed 10GB. For performance reasons, smaller databases are more manageable. This is a Lotus Domino Document Manager-specific recommendation.

When setting up a file cabinet, administrators can choose whether to have Lotus Domino Document Manager automatically create a new document database once a specified threshold has been exceeded. Keep in mind, however, that with more document databases, creating a full-text index for these databases increases their size by approximately one-third.

Number of libraries
A single library may prove adequate for an implementation of Lotus Domino Document Manager. However, administrators can set up multiple libraries on the same server. For example, a separate library can be created for each division in a company. There can be one or more library databases with separate security groups on a single server. Using multiple libraries is a good way to distribute data and accomplish some user load balancing.

Number of file cabinets
File cabinets consist of at least two databases: a binder database and a document database. These are the databases that will increase the most in size as more data is added to the system. Consequently, file cabinets use the most system resources.

Number of binders
Theoretically, there are no limits to the number of binders that can be stored in a file cabinet. However, there are some practical limitations to note when accessing a file cabinet that contains thousands of binders.

For navigation performance reasons, the number of binders in a single file cabinet database should not exceed 2,500 for all clients (Notes, web browser, or other). The number of binders displayed in the Site Map (in the web browser) should not exceed 1,000. Otherwise, the “Next” and “Previous” links appear and users may find it impractical to navigate in the structure if an additional click is required to see all the binders. Any number greater than 2,500 has a direct negative impact on performance of the entire library. For example, opening a file cabinet that contains 50 binders immediately after opening one that contains more than 2,500 binders takes about the same amount of time to display in the Site Map because nothing is cached.

Number of documents
The total number of documents in a single file cabinet should not exceed 250,000. Documents with multiple drafts or versions should be factored into this number as well. Databases with fewer documents perform better. This is because scheduled agents (such
as the DocumentMonitor agent) that run on all documents can prevent other agents from running if the scheduled agents have not completed.

**Number of document types**
To avoid end user confusion, the recommended maximum number of document types per file cabinet is 30.

**Document size**
Lotus Domino Document Manager treats all documents the same, regardless of size. For this reason, a document database that contains 1,000 50MB documents performs better than the same database when it contains 10,000 5KB documents.

**Version control**
All document versions are stored in the document database unless they are archived. If the number of document versions increases significantly, consider implementing triggers that would archive older versions to less expensive media. The growing document chain can reduce performance and hence it is recommended to encourage users to delete drafts when a new version is created. This will reduce the total number of documents in the database.

**Anticipated document growth**
Consider all the sources that documents will be coming from: migration from legacy systems, file systems, Notes databases, e-mail, new documents, scanning, and so on. Rules and policies should be set for the various document types to ensure that they are managed, controlled, retained, and disposed of accordingly. For example, subscription notices can be used to ask authors to update, delete, or archive documents after a certain number of days, as a way to keep information up to date.

**Library Considerations**

**Why use replication?**
Replication allows information to be stored and processed on different servers, while the data is synchronized over time. Replication offers two benefits. First, it ensures that Lotus Domino Document Manager is highly scalable, not only vertically but also horizontally. Rather than upgrading to expensive machines to gain improved performance, you can add less expensive machines to share or split the work. This type of upgrade also helps minimize disruption to users. Secondly, replication lets you distribute work in different locations. This allows users to access data from the server that is closest to them, and thus reduces unnecessary network traffic. For more information about common replication topology strategies, see Chapter 6 of the *Lotus Domino Document Manager Administrator’s Guide*.

**File Cabinet Considerations**

**Which is better, ActiveX or Folders Table of Contents (TOC) design?**
Some benchmarking tests that compare performance when opening file cabinets containing binders with 25,000 documents show that the ActiveX design yields better performance than one that uses the Folders TOC design. The reason for this is that a file
cabinet using the Folders TOC design contains more views than one that does not. Unless there is a business reason that requires using the Folders TOC design (for example, the binder profile contains required fields that must be filled in), it is recommended to use the ActiveX design. Note that selecting the ActiveX design prevents the use of custom binder views.

**What about setting binder and document thresholds?**

Administrators may want to set binder and document thresholds as a way to distribute binders and documents across multiple databases, and ensure that performance remains acceptable. Document size has little impact when opening a document database. However, the number of documents in a database does impact performance and should be taken into consideration when setting the threshold.

The BinderThreshold and DocumentThreshold agents run at a scheduled time to spawn a new database if threshold criteria (set in the file cabinet profile) has been met. The agents create the new database as a copy of the original database, and then increment the number next to the new database name.

**What is the purpose of cloning and moving file cabinets?**

The “Clone File Cabinet” feature lets administrators copy an existing file cabinet, including its binder structure. This clone can subsequently be moved to a different library.

The “Move File Cabinet” feature lets administrators move a file cabinet from one library to another within Lotus Domino Document Manager. This allows for greater flexibility throughout the life cycle of a file cabinet. For example, it can be used to move file cabinets from test to production domains, to reorganize libraries, or as a form of archiving.

**Binder Considerations**

**How and when to use the Binder Administration functionality?**

Administrators may need to merge all the documents for a cabinet together if there are multiple document databases. The best approach is to use the “Move Binders to a Different Document Database” button using the Binder Administration functionality. The binders themselves are not moved but the binder is set to use a different document database. All documents currently in the binder are moved to the new document database and marked for deletion from the current database. Multiple binders can be moved at the same time. This functionality is available only in the Notes Client on the master replica of the file cabinet, and for use by File Cabinet managers and System Administrators.

Binder restrictions and special considerations when moving binders:

- The binder cannot be checked out to any user. Attempting to do so will result in an error message.
- The binder cannot contain any documents that are checked out on a non-master replica. Attempting to do so will result in an error message.
- All versions, drafts, and working copies of each document in the binder must be moved.
- All review copies and discussion documents must be moved.
- Favorites, recently edited links, checked out document links, and life cycle status links must all be updated to point to the new document database.

**Creating document databases.** There is no relationship between manually created document databases and the thresholds that are defined for the file cabinet. However, the file cabinet must be configured to use multiple databases.

**Deleting unused document databases.** Document databases cannot be deleted unless they are empty and are not used by any binders.

**Displaying document database information.** The administrator can view a list of all document databases for a particular file cabinet, and display additional information for a selected database. This information includes the path, replica ID, size, and whether the database has a full-text index.

**Displaying binder information.** The administrator can view a list of binders categorized by document database, to see the distribution of binders across document databases. Additional information about a particular binder can also be displayed. This includes the number of documents, in addition to the document database information listed above.

**Document Considerations**

**Is there a good naming convention and categorization method for documents?**

It is good practice to name and categorize documents according to the file cabinet in which they belong. Duplicate names, although allowed, are not recommended, particularly when using the Lotus Domino Document Manager API, to get a handle on documents by title. Similar file and document types should be stored in the same file cabinet so that macro policies such as retention, archival, and disposition can be applied on a per file cabinet basis.

**Is it possible to program a document naming convention into Lotus Domino Document Manager?**

It is possible to program a document naming convention along the lines of xxx-xx-xxx.

There are several ways to do this. You can:

- Add a computed field to each Document Type subform.
- Add a computed field to the New Document and Document forms (note that this presents a future upgrade issue).
- Add the field via the QueryCheckIn document event for new documents.

When using the QueryCheckin event, it is possible to add a sequence number script; this allows greater flexibility than formulas, because there is a different sequence for each file
cabinet or binder. The fields can be changed using the doc object of the event.

**What kinds of file attachments can be full-text indexed?**
Full-text indexing is handled by the indexing engine native to Domino. Check the documentation for the Domino version you are using for a list of documents that can be indexed.

**Creating hyper links to documents in Lotus Domino Document Manager**
Lotus Domino Document Manager now allows users to create URL links to documents. The URL can be a static URL, *i.e.*, to the specific version of a document, or it can be a dynamic URL, *i.e.*, it will reference the latest version of a document.

This feature is available from the browser interface and to get to it, users must select the Forward action from the Document Menu. Selecting the “Send File As Link” checkbox will give you the option to create a static or dynamic link.

**Desktop Enabler Considerations**
The Desktop Enabler allows easy access to Lotus Domino Document Manager from the user desktop.

**The install process**
Since there is no batch-user install for the Desktop Enabler, users can install the Desktop Enabler manually from the Lotus Domino Document Manager user interface or administrators can use a third party install tool such as Novell® ZENworks® to do this.

**Accessing the library**
The first time a user accesses the Lotus Domino Document Manager library with the Desktop Enabler, they must know the URL for it. Using a centralized install tool allows the Administrator to set the library for all the users.

**Document types supported by the Desktop Enabler**
The Desktop Enabler is designed to work with all ODMA-enabled applications. ODMA is a systems protocol that allows desktop applications such as Microsoft Word and Lotus Word Pro® to communicate with document management applications such as Lotus Domino Document Manager. Configuring the Desktop Enabler settings allows users to set what ODMA application to use with Lotus Domino Document Manager.

**Lotus Domino Document Manager Office Dialogs**
The Desktop Enabler now integrates with Microsoft Office using Microsoft Office Dialogs. This allows Lotus Domino Document Manager to work with Microsoft Word, Microsoft Excel and Microsoft Visio and to exchange metadata with those applications. Metadata such as the Revision, Document ID, Binder, File Cabinet and Library can be exchanged and stored in the document properties of the Microsoft Office document.

**Search Considerations**
Domain Search
Starting with release 6.5.1, Lotus Domino Document Manager provides the option to use Lotus Domino Domain Search to perform searches across multiple libraries. Using Lotus Domino Domain Search is a much faster way of searching across Lotus Domino Document Manager databases because a single index is created of all the documents. Searching occurs against this single index instead of searching through multiple full-text indexes of the file cabinet databases returns search results much faster.

Lotus Domino Domain Search can be used for all clients connecting via HTTP including the Desktop Enabler. It should be noted that even if Lotus Domino Domain Search is enabled for the library, Lotus Notes client users, and Desktop Enabler users who connect via NOTES protocol, are still required to use the standard full text search (a full text index is still created for every file cabinet database). Lotus Domino Domain search is not available for these clients.

When creating, or modifying a file cabinet, the administrator has the option to include it in the domain search.

With Lotus Domino Domain Search enabled, there are a few new fields added to the Advanced Search form and these are:
- Search Word Variants
- Fuzzy search
- Use thesaurus synonyms

For smaller installations, the domain search server may be the Lotus Domino Document Manager server, but a separate stand-alone domain search server is strongly recommended for larger installations.

Single sign-on must be enabled if the domain search server is on a separate server. Single sign-on is required so web client users don’t have to re-authenticate when performing a search. It is also required for Desktop Enabler users, so they can perform a domain search. The DIIOP server task must be running on the domain search server because the Lotus Domino Document Management server connects to the domain search server in this way.

Limitations of the Domain Search
Index updates occur only as often as specified in the domain search server’s server document. This means that domain search will return information that is only as current as the time of the most recent index update. When a document’s content changes, the domain search index will be slightly out-of-date until the next index update occurs. This can cause domain search results to report documents that have been modified or have changed states. For example if a new version or draft of a document is created, domain search results will indicate a previous version of the document as being the latest, and will be unaware of the new version until the next index update occurs. The schedule is typically done daily, during late night hours, but it should be done hourly for a dedicated domain search server.
**Ignoring certain file types in the full-text indexer**
Administrators can choose to exclude various file types from the Domain Index. For example, file types such as .tif and .jpg can consume considerable amounts of space. To ignore these file types, add the following line to the notes.ini file on the server (using an asterisk and a period before the file type name and a comma to separate multiple file types), and then restart the server:

```plaintext
FT_INDEX_IGNORE_ATTACHMENTTYPES=*.tif,*.jpg
```

**Increasing the size of the full text indexer**
Administrators can increase the default size of the full-text index by setting the following parameter in the notes.ini file: `FTG_INDEX_LIMIT= X` with the value of X calculated as follows: 8MB is calculated as 8*1024 *1024=8388608; thus, `FTG_INDEX_LIMIT=8388608` 12MB is calculated as 12*1024*1024=12582912; thus, `FTG_INDEX_LIMIT=12582912` 

**Note:** There is no limit to the size set on `FTG_INDEX_LIMIT`. This limit is for the entire server and is not for each individual full-text index.

**Customization Guidelines**
When customizing Lotus Domino Document Manager, it is recommended that you use a tool such as Teamstudio® CIAO distributed by Ives Development Inc. or TS Analyser to track all changes. Using change control has the following benefits:

- Ensures that all custom Lotus Domino Document Manager templates can be checked in and modifications documented.
- Ensures that templates will be versioned when ready for deployment to a production environment, and tracks changes to the templates.
- Allows developers to revert easily to previous templates if needed.
- Simplifies future Lotus Domino Document Manager upgrades because all template changes are documented.

**Note:** An upgrade of Lotus Domino Document Manager will lay down new templates. If there are any customizations made directly in the Lotus Domino Document Manager templates, the templates are moved to the "ddbu" back up subdirectory during upgrade. If the customizations are made in a copy of a Lotus Domino Document Manager template, and has a distinct template design name different from the standard Lotus Domino Document Manager template, this type of customized template is not touched during upgrade. For this reason, it is strongly recommended that you document all template changes. Reapply any customizations to the new templates or copies of the new templates after upgrading.

**Guidelines for agents, views, and script libraries**

*Views*
Lotus Domino Document Manager makes use of many views, and hence as the number of documents increase, performance may slow down. It is possible to add custom views
to a Lotus Domino Document Manager system; the recommended maximum number is 75. Adding more than 75 views may decrease performance.

Two ways of reducing negative performance impact are to reduce the number of sortable columns in your view, and to reduce the amount of data being sent down to the client. This data could be meta-data or code. In the case of data, keep meta-data subforms small. However, if you plan on using meta-data to sort or for categories, do not build them on the fly. Create another field and pre-populate sort keys and categories. Forcing the view to render these on the fly is time consuming. Also, try to reduce the over all number of characters being sent to down to the client. For browser specific JavaScript functions, remove white space and comments. If building custom routines, store the JavaScript functions in a static frame on the WebHeaderPanel form.

**Agents**
Lotus Domino Document Manager makes use of agents throughout the product. While it might be necessary to incorporate additional agents to perform particular tasks, this could have a direct negative impact on performance, since system resources can be tied up by running an agent. Reusing code via script libraries is generally more efficient than using agents.

**Script Libraries**
To streamline event processing it is recommended to delete all “blank” code from the view and form events of the library and file cabinet templates. There are times when a developer will go into an event and create an empty Sub XXX...End Sub block in an event. Initially, the impact is negligible but can begin to impact performance as additional similar events occur in sequence.

**Guidelines for forms, subforms, and fields**
- There should be no more than 80 fields on a document subform. A document with fewer profile fields yields better.
- Rich text fields for meta-data are not supported in all clients and should not be used.
- Some mechanism should be built into each subform to ensure that fields are saved upon form refresh.
- Developers who include hidden fields in binder/document subforms that are intended for use with the ODMA client should use a prefix notation (for example _Fieldname1) to ensure that the hidden fields will sort to the bottom of the ODMA display.

**Pre-deployment checklist**

26
Before you deploy custom templates, make sure you complete the following tasks for each template:

1. Check the Access Control Lists (ACLs) to make sure they are correct. It is a good idea to compare them with the original Template.
2. Check the template properties to make sure the template does not inherit its design from other templates.
3. Recompile all Script Libraries that have a dependency on a Script Library that was changed. All design elements that depend on that Script Library also need to be recompiled. (To recompile a design element, open the element, add a space, delete the space, and then save).
4. Sign the template with the Administrator ID.
5. Open the template in Domino Designer, then check each design element as follows: First, verify that the “Last Modified by” field lists the administrator. Next, select the design element, and then choose Design Properties from the right-mouse button menu. Click the Design tab in the InfoBox and verify that “Inherit from the design template” is blank. Be sure to check each design element in Forms, Views, Agents, Subforms, Script Libraries, Database scripts, and Navigators.
6. Check that the template has been versioned in source control. If you need to recompile, make sure the template is checked in with changes and reversion.

**Phase 3: Staging**
The objective of this phase is to test the Lotus Domino Document Manager deployment with real users and real data, and collect feedback prior to the Production phase. This is the ideal time to pilot the system with a few users. It is good practice to incorporate some kind of feedback mechanism into the system. Users should have a way to provide feedback, including whether the system simplifies their work and is intuitive enough to use. Users should also be able to report any bugs they encounter. To ensure consistency, it is important to set up the development, staging, and production server identically so that they have the same operating system and Domino version.

**Performance Tuning Tips**
During the Staging Phase, it is recommended to tune the system so as to get optimal performance. Below are some general areas that if modified, will improve performance on your server.

**Paging Space**
The OS recommendation for swap file size should be respected: i.e. it must be equal to or greater than the amount of RAM on the system. It is also recommended that the paging space be configured across multiple disks to maximize performance.

**Notes.ini settings**
Domino Versions 5.0.12 and above have many stability and performance improvements for applications running on Domino. Add the following notes.ini settings:

- **ENABLE_LOCAL_PCB=1** (This enables a performance fix available in 5.0.12+)
- **SERVER_NAME_LOOKUP_NOUPDATE=1** (Allows authentication to take place while rebuilding NAB views)
- **SERVER_MAX_CONCURRENT_TRANS=80** (Allows more concurrent transactions for the server (only useful for Notes clients))
- **FTG_USE_SYS_MEMORY=1** (Allows FTI to allocate memory directly from the heap)

**Enabling concurrent web agents**

Enabling "Run web agents concurrently?" in the server document allows multiple agents to be executed simultaneously when requested via HTTP.

**Full text indexing**

FTI is a very intensive operation and it performs better with batch updates. Configure the update task to run every 15-60 minutes.

This task is not necessary if you are using Domain Search.

**Adding AIXTHREAD_SCOPE=S in the Unix environment**

AIXTHREAD_SCOPE={P|S}, where P signifies process based contention scope and S signifies system based contention scope. In general Domino performs better with system contention scope.

**Modifying Filecab.ntf Agents**

**ScheduledAgentsDocsCreateorMod**

**Description**

This agent runs if documents have been Created or Modified.

It contains the following agents:

1. **RetrievefromBRMS** – This agent is used to retrieve archived documents on an iSeries server
2. **ProcessApprovedDocuments** – This agent is used to change the status of any documents that are pending approval to “Approved State”
3. **DelayedFolderCreate** – This agent is used to create the binder table of contents for new binders
4. **FullTextIndexer** – This agent is used to update the full text index of each file cabinet database
5. **AA_Summarize** – This agent is used to populate the document summary field on the document profile with the first 150 characters of the file attachment

**Modifications**
1. Disable RetrievefromBRMS if archiving is not being used
2. Disable FullTextIndexer agent. The Domino server updates the full text indexes every night, which may be sufficient.

**DocumentMonitor**

**Description**
This agent is used to make sure security is correct on documents and binders and is set to run once per hour.

**Modifications**
Modify it to run once a day, as opposed to once an hour. This is done because it may slow down system performance, since it runs on almost every document, every hour. It is possible that document monitor would never stop running if there are many documents in each file cabinet.

**ScheduledAgents2am**

**Description**
This agent is used to run several scheduled agents and runs on all documents
It contains the following agents:
1. **MarkforArchive** – This agent is used to flag documents to be archived
2. **ProcessExpiredReviews** – This agent is used to send a reminder e-mail to a document reviewer if the time to review a document has expired
3. **RefreshBookmarks** – This agent is used to update bookmarks pointing to documents if they have been updated

**Modifications**
Disable this agent if you are not using the archiving or review and approval features.

**ScheduledAgents3am**

**Description**
This agent is used to run several scheduled agents and runs on all documents
It contains the following agents:
1. **ArchivetoFilesystem** – This is a sample agent used to archive documents to a file system
2. **ArchivetoBRMS** – This is a sample agent used to archive documents to the file system on an iSeries.
**Phase 4: Production**
Before attempting to migrate from the staging server to a production server, it is extremely important that you back up all data. When you are ready to migrate the server, you can do so in several ways.

Technote #182465 *How to Migrate Lotus Domino Document Manager* from one site to another contains detailed instructions that step you through migration.

You can also use a third-party product such as Norton® Ghost® distributed by Symantec Corp. to make an image of the staging server that can then be placed on the production server. An alternative (but not necessarily faster) method is to perform a file system copy to the production server. In either case, administrators will need to reconfigure the production server to match the staging server’s IP address. In addition, the production server’s registry will need to include the appropriate Lotus Domino Document Manager entries. To update the production server’s registry, install Lotus Domino Document Manager on the production server prior to performing the file system copy.

**Lessons Learned**
-When setting up a Lotus Domino Document Manager replica server, the library and file cabinet templates (domdoc.ntf and filecab.ntf) must be placed in the root of Domino (i.e. /Lotus/Domino/Data) on the replica server. Once the Lotus Domino Document Manager replica server is set up, the files can be moved, if desired.

-When cloning a file cabinet for connection to a library, the user who will connect the file cabinet must be explicitly defined by name in the file cabinet security. Membership in a group is not sufficient. This is a known limitation.

-The default binder cannot be removed from a file cabinet, nor can the binder security be modified. It can only inherit the file cabinet security. The default binder is best left unused, especially if custom binder security is used. If binder categories are used, the default binder will sort to the bottom of the category list (under Not Categorized). To hide the default binder, simply use an `@If(Title="Default";...` condition in the view selection formula.

-When creating file cabinets on a given Lotus Domino Document Manager server, only one user at a time should do so. Otherwise, multiple file cabinets may be created and assigned the same database file name.

-Do not use the forward slash character (/) when naming groups. Domino uses this character as a hierarchical name separator. For example, if you name a group
-Moving large amounts of data into the system through batch scanning or migration tools should be done on a different server from the production server. Once complete, this data can then be replicated to the production server. This ensures that the production server resources remain available for end users. If this server is in the same domain as the production server, it can be set up as a library replica server (see Chapter 6 of the *Lotus Domino Document Manager Administrator’s Guide* for details). If this server is in a different domain than the production server, Technote #180280 Instructions for Setting Up Cross-Domain Replication for Lotus Domino Document Manager provides details on how to configure the servers.

**References**

- [http://www.domino-guru.co.uk](http://www.domino-guru.co.uk)
- Lotus Domino Document Manager 6.5.1 Readme.txt
- RAID Without Additional Hardware: Do It Yourself With Windows 2000
- RAID brings reliability to mass storage
  [http://techupdate.zdnet.com/techupdate/stories/main/0,14179,2815047,00.html](http://techupdate.zdnet.com/techupdate/stories/main/0,14179,2815047,00.html)
- IBM Services for Storage Area Networks - An Overview
- Designing an IBM Storage Area Network – Redbook
- Technote #180280 *Instructions for Setting Up Cross-Domain Replication for Lotus Domino Document Manager*
The information contained in this publication is provided for informational purposes only. While efforts were made to verify the completeness and accuracy of the information contained in this publication, it is provided AS IS without warranty of any kind, express or implied. In addition, this information is based on IBM’s current product plans and strategy, which are subject to change by IBM without notice. IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, this publication or any other materials. Nothing contained in this publication is intended to, nor shall have the effect of, creating any warranties or representations from IBM or its suppliers or licensors, or altering the terms and conditions of the applicable license agreement governing the use of IBM software.

References in this publication to IBM products, programs, or services do not imply that they will be available in all countries in which IBM operates. Product release dates and/or capabilities referenced in this publication may change at any time at IBM’s sole discretion based on market opportunities or other factors, and are not intended to be a commitment to future product or feature availability in any way.

IBM, the IBM logo, Domino, Lotus, Notes, Sametime, SmartSuite, and Workplace are trademarks of International Business Machines Corporation in the United States, other countries, or both. Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

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

GC28-7791-00