Lotus

White Paper

Domino.Doc Best Practices

January 2002

A Lotus Software White Paper

IBM®

© Copyright 2002 IBM Software Group. Lotus Software, IBM Software Group, One Rogers Street, Cambridge, MA 02142. Domino is a trademark, and Lotus, 1-2-3, Lotus Notes, Notes, Domino.Doc, Freelance Graphics, and Word Pro are registered trademarks of Lotus Software or IBM Corporation, or both.

IBM, AIX, OS/400, OS/390, and Tivoli are registered trademarks of International Business Machines Corporation, used under license.

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, and service names may be trademarks or service marks of others.

Contents

Introduction

There is no one-size-fits-all strategy for deploying Domino.Doc®. The IBM® Software Group can offer many recommendations and suggestions, as detailed in this document. However, effective deployment of Domino.Doc 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 Domino.Doc successfully also requires the unique skills of an individual who can understand the business requirements and apply them to the document management system.

The process of deploying Domino.Doc typically consists of four phases: Planning, Development, Staging, and Production. Each phase has a clearly defined objective. This white paper 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 Domino.Doc. 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 *Domino.Doc Administrator's Guide*, and *Domino.Doc Programmer's Guide*. When installing or upgrading Domino.Doc, follow the procedure listed in the *Domino.Doc Installation Guide*. These resources contain additional useful information that is not included below.

This white paper is written for the Domino.Doc 3.0 and 3.1 releases.

Acknowledgments

The author would like to thank the Advanced Collaboration Group SWAT Team, the Domino.Doc Development Team, Lotus Professional Services, Lotus Solutions, and the KNAC Team for contributing content and reviewing this white paper. Special thanks goes out to our customers who let us gather data from their Domino.Doc deployments, the Domino.Doc Business Partner community, in particular, Advanced Technology Consulting, Cobra Technologies, Correlate Technologies, Hadleigh Marshall International Ltd. and International Technology Alliance for their contribution and support, and for all those who participated in the public discussion forums on Domino.Doc and provide valuable insight and feedback.

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 the 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 Domino.Doc to manage their documents?

Notes domains

- Are there multiple Notes® domains?
- Do the 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 Domino.Doc hierarchy?
- How are users going to access the system (for example, via the Web, a 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 Domino.Doc?
- Are these applications Open Document Management API (ODMA)-enabled?

Hardware Selection Guidelines

Because Domino.Doc is system resource-intensive, capacity planning should be given careful consideration. The hardware on which Domino.Doc 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.

Some performance tests have shown that the more processors a machine has, the faster it performs end user tasks such as creating and opening documents. For example, performance improved by up to 10 seconds in these areas when comparing a six-processor machine with a four-processor machine.

To ensure high performance in Domino.Doc 3.0 and 3.1, each processor should be at least 500Mhz. When purchasing your server, it is recommended that you choose the fastest processor available.

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. A good article to obtain more information on this is "RAID brings reliability to mass storage" available online at http://techupdate.zdnet.com/techupdate/stories/main/0,14179,2815047,00.html.

It is recommended that you use a RAID 1 disk array for Domino™ databases, and an expandable RAID 5 array for Domino.Doc databases spread across the 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 Domino.Doc 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 it is recommended to read the IBM redbook, "Designing an IBM Storage Area Network" available online at http://www.redbooks.ibm.com/redbooks/SG245758.html.

Network and bandwidth

Domino.Doc requires high bandwidth for maximum performance. Where possible, try to use 10Mbps cards and networks scalable to 100Mbps, and use multiple LAN segments (one for each partition) to isolate network traffic at high end user loads. TCP/IP is a scalable protocol and is recommended for use with Domino.Doc. Over slower connections, it is recommended that you set up Domino.Doc 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.

When accessing Domino.Doc, response time is much faster via HTTP protocol through a thin client (such as a web browser) than through a thick client (such as Notes); this is due to the amount of data sent between the server and the client for processing. Thin clients do all the data processing on the server and just display the results in the browser, while thick clients require some processing of the data to be done on the server and some on the client.

Using the Domino.Doc Notes client over a bandwidth of less than 1Mbps is not recommended because of exponentially increased response time. Protocols such as SPX and NetBIOS are not recommended because they are slower. If these protocols are not being used, it is best to disable these ports on the server as well as on the Lotus Notes client machines.

Availability

High server availability is necessary, especially in global deployments of Domino.Doc, and should be 24/7 if possible. This can currently be achieved through OS level clustering. At the time of writing this white paper, Domino.Doc running on a Microsoft Cluster Server (MSCS) is the only OS level clustering system tested and supported by the Domino.Doc Development Team. Administrators should note that only one Domino.Doc cluster should be up at a time in a clustered system. Domino clustering is not supported.

When choosing your hardware, select a server that has additional failover on every hardware component (for example, 3 power supplies, 2 network cards, alternative network paths, and so on). Several high-end servers currently available on the market have standard hardware component failover.

Backup and recovery

It is good practice to schedule backups on an incremental regular basis. All system data should be backed up at least once weekly, and high priority data should be backed up daily.

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.

Operating System Considerations

The Domino.Doc server can run on the following platforms: Windows NT 4.0, Windows 2000, AIX®, Solaris, OS/400®, and OS/390®. Each platform has benefits that may vary depending on existing IT policies and administrators' skills sets.

Using Windows NT/Windows 2000

Windows NT/ Windows 2000 is good for small to medium-sized installations of Domino.Doc. Typically, this platform does well in environments where the number of users is less than 2000, and the amount of data is less than 300GB. Running Domino.Doc on the Windows platforms lets you use the Domino.Doc API, an Object Linking and Embedding (OLE) Automation Server and its methods and properties. The Windows environment is the only one that lets you use the Domino.Doc API to run agents for processing data on the server. This is the only platform on which Domino.Doc supports OS clustering.

A disadvantage to using Windows NT 4.0 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 good for small to large installations of Domino.Doc. UNIX is a good platform for companies that may be running several Domino applications in addition to Domino.Doc on the same server.

A disadvantage to using the UNIX platform is that it is difficult to integrate Domino.Doc with other applications because the Domino.Doc API is not available on UNIX.

Using OS/400

IBM OS/400 is a scalable, stable, and high performing platform that is good for medium to large installations of Domino.Doc. The iSeries system scales well as you can have up to 12 processors running, use up to 40GB 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. Domino.Doc for iSeries provides specialized agents to archive and retrieve documents using Hierarchical Storage Management (HSM).

The disadvantages to using OS/400 are that the hardware is expensive, the Domino.Doc API is not available on OS/400, and the iSeries version for Domino.Doc does not support Double-Byte Character Set (DBCS) file names on Internet Explorer 5 (International).

Using OS/390

Running Domino.Doc on a mainframe system such as IBM S/390 is advantageous because it is a powerful system with high availability, high performance, high reliability, and high security. Mainframes have very sophisticated system management, backup, and restoration tools that help system administrators allocate and manage system resources.

The disadvantages to using OS/390 are the cost of the hardware, which can be quite expensive, and not being able to use the Domino.Doc API.

Phase 2: Development

The main objective of the Development phase is to configure and customize Domino.Doc 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 of this phase is to tune server performance. Commercially available performance measuring tools such as Segue Silk Performer or QALoad from Compuware can be used to simulate user behavior and to load data to create performance benchmarks.

Architecture

Different companies have different uses for Domino.Doc. 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 1) 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 s/he 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 implications of using the modular architecture model are 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.

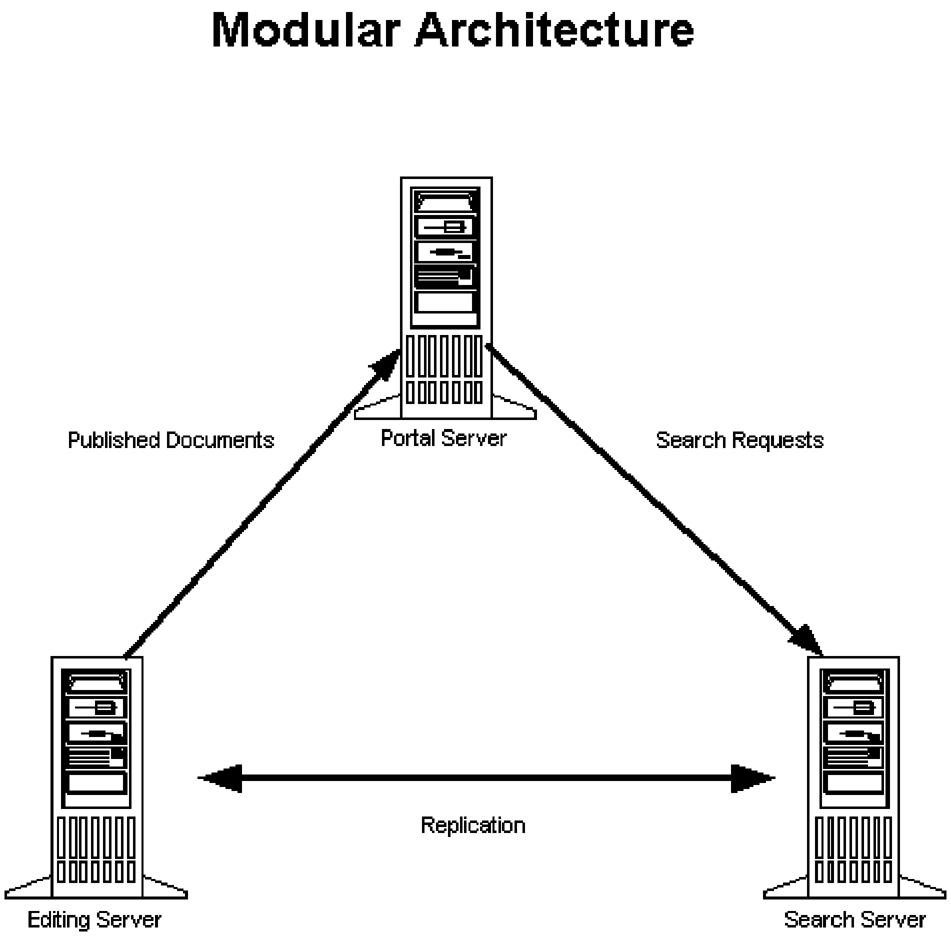


Figure 1

Volumetric Analysis

Database size

The maximum size of each Domino.Doc database should not exceed 10GB. For performance reasons, smaller databases are easier and faster to work with.

When setting up a file cabinet, administrators can choose whether to have Domino.Doc 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. This size increase will have a direct negative impact on search performance.

Number of libraries

A single library may prove adequate for an implementation of Domino.Doc. However, administrators can set up multiple libraries on the same master 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 do 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 more system resources than any other Domino.Doc databases.

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 (located 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 confusion, the recommended maximum number of document types per file cabinet is 30.

Document size

Domino.Doc 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 the document chain reduces performance and hence it is recommended to encourag 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 Domino.Doc 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 *Domino.Doc Administrator's Guide*.

File Cabinet Considerations

Which is better, ActiveX or Folders 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 a nearly empty cabinet 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. 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 Domino.Doc. 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 Domino.Doc 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 Domino.Doc?

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

If the application used to generate a document does not store strings in ASCII format, the document cannot be indexed because the indexer extracts only ASCII strings from file attachments. A partial list of application file types and whether they support full-text indexing is shown below in the following table.

File type	*Can this be indexed?*
Word Pro® 96 (lwp)	No
Word Pro 97/9.5 (lwp)	Yes
Word Pro 97 (small file format)	No
Freelance Graphics 97 (flg)	No
Freelance Graphics 9.5 (flg)	Yes
Word 97/2000 (doc)	Yes
Adobe PDF (pdf)	Yes
1-2-3® (wk4)	Yes
Quattro Pro 7	Yes
ASCII text file (txt)	Yes
PowerPoint (ppt)	Yes
WordPerfect 6.0 (doc)	No
Correlate K-Map (kma)	Yes

WordPerfect 6.0 compresses the data in the files.

Word Pro 97 has a “Small file format” user preference which is a compressed format. To index Word Pro 97 documents, turn off the “Small file format” option. Word Pro 96 always stores the document in compressed format; therefore, it cannot be indexed.

Freelance Graphics 97 stores strings in a double-byte format; therefore, these documents cannot be indexed.

Many applications store control information in ASCII format. Control information is used to determine how the data that the user enters is stored and displayed. Such information is generally not visible when documents are viewed and only appears in a file that the application generates when saving. Any ASCII control string is extracted by the indexer. Thus applications that store ASCII control information may generate false positives during a search.

There is no full-text indexing capability of binary attachments (Adobe PDF files, Microsoft Word files, and so on) on the OS/400 platform. Domino for OS/400 supports indexing of attachments for single-byte plain 7-bit ASCII text files only. Other file types may produce inaccurate results.

Creating hyper links to documents in Domino.Doc

There are several ways to create a hyper link to a Domino.Doc document. One example can be found in the Domino.Doc samples database (available for download at http://www.lotus.com/dominodoc).

Another way to create a hyperlink is to create an API agent that, when handed a persistent document ID, opens that document in a browser window. A subform can be created in a non-Domino.Doc database to allow a user to navigate in a library through file cabinets and binders to select a document by title and then to store the URL. If the administrator wants to disregard security and allow anyone to click on the URL to open the document, the agent can be placed in the Library template. Alternatively, if the administrator wants to maintain security and force a Domino.Doc login to view the document, the agent can be placed in the File Cabinet template.

Desktop Enabler Considerations

The Desktop Enabler allows easy access to Domino.Doc 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 Domino.Doc user interface or administrators can use a third party install tool like Novell ZENworks to do this.

Accessing the library

The first time a user accesses the Domino.Doc library with the Desktop Enabler, they must know the URL for it. Using a centralized install tool (like Novell ZENworks) allows the Administrator to set the library for all the users.

Creating a Library link

If Domino.Doc is not installed in the root of the Domino Data directory, administrators will need to create a library link to work with the Desktop Enabler and the web browser. If this link is not created, or used for Desktop Enabler access or Web browser access, the users will get multiple login prompts for every binder they access. Technote #164392 Domino.Doc Prompts for Web Authentication More Than Once When Installed to Subdirectory details the steps to create this link.

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 to communicate with document management applications such as Domino.Doc. However, not all applications are ODMA enabled. A complete list of ODMA enabled applications is available at http://www.infonuovo.com/odma/faq/ODMAapps.htm. Configuring the Desktop Enabler settings allows users to set what ODMA application to use with Domino.Doc.

Common applications not ODMA enabled

The most common applications that do not support the ODMA protocol are Lotus 1-2-3, Microsoft Excel (all versions), and Microsoft PowerPoint 97 (although Microsoft PowerPoint 2000 supports ODMA).

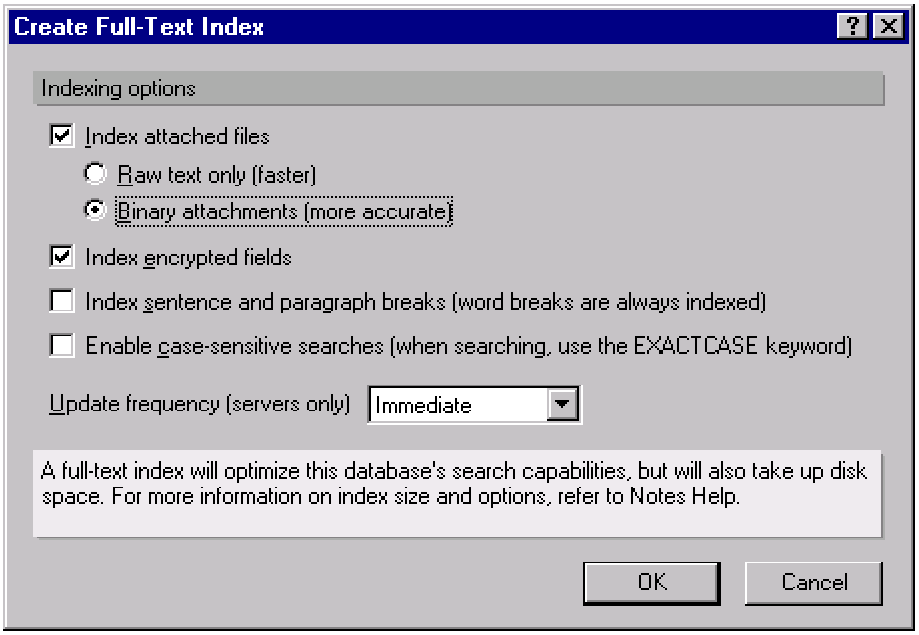
To enable these applications to work with Domino.Doc, macro tools are available for download at http://www.lotus.com/dominodoc.

Search Considerations

The speed of search in Domino.Doc is directly related to the number of databases in a file cabinet. If the file cabinet contains more than 3 databases, there is a negative impact on search performance. For example, in performance benchmarking tests that compared a 10GB file cabinet containing six document databases with a 10GB file cabinet containing one database, search times were 4-5 times slower in the file cabinet containing the six databases than in the file cabinet containing the single database. The reason for the decrease in response time is when the number of document databases increases, search in a file cabinet will sequentially search in all document databases, one by one. The total time it takes is the time it would take on one database multiplied by the number of document databases for this file cabinet.

Enabling full-text indexing on binary attachments

In Domino versions 5.0.5 or higher, the GTR indexing engine can index .pdf files. To do this, open the document database using the Notes Administrator client. Select the Database Properties option from the File menu. In the Index Options tab, click on the Create Index button. In the “Create Full-Text Index dialog box,” turn on “Index attached files” and select “Binary attachments (more accurate).” Note: This option is not available via the file cabinet properties in Domino.Doc.



Indexing options

☑ Index attached files

○ Raw text only (faster)

● Binary attachments (more accurate)

☑ Index encrypted fields

□ Index sentence and paragraph breaks (word breaks are always indexed)

□ Enable case-sensitive searches (when searching, use the EXACTCASE keyword)

Update frequency (servers only) Immediate

A full-text index will optimize this database's search capabilities, but will also take up disk space. For more information on index size and options, refer to Notes Help.

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:

```
FT_INDEX_IGNORE_ATTACHMENT_TYPES=*.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 Domino.Doc, it is recommended that you use a tool such as Team Studio's CIAO from Ives Development Inc. or TS Analyser to track all changes. Using change control has the following benefits:

- Ensures that all custom Domino.Doc 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 Domino.Doc upgrades because all template changes are documented.

Note: An upgrade of Domino.Doc will lay down new templates. If there are any customizations made directly in the Domino.Doc templates, the templates are moved to the "ddbu" back up subdirectory during upgrade. If the customizations are made in a copy of a Domino.Doc template, and has a distinct template design name different from the standard Domino.Doc 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.

Pre-deployment checklist

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

1. Check the Access Control List (ACL) 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 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.

Guidelines for agents, views, and script libraries

Views

Domino.Doc makes use of many views, and hence as the number of documents increase, performance slows down. It is possible to add custom views to a Domino.Doc system; the recommended maximum number is 75. Adding more than 75 views decreases performance in the databases.

Two main ways of reducing the performance hit 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

Domino.Doc is an agent-heavy 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 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 performance than a document with many.

Rich text fields are not supported and should not be used.

The size of a field should not exceed 64K. This is a known Notes limitation.

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.

Phase 3: Staging

The objective of this phase is to test the Domino.Doc 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.

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 Domino.Doc from one site to another contains detailed instructions that step you through migration.

You can also use Norton Ghost 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 Domino.Doc entries. To update the production server's registry, install Domino.Doc on the production server prior to performing the file system copy.

Lessons Learned

When setting up a Domino.Doc replica server, the library and file cabinet templates (domdoc.ntf and filecab.ntf) must be placed in the root of Domino on the replica server. Once the Domino.Doc 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.

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 Domino.Doc 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 Officer/Person, Domino interprets Officer as the group name, and Person as the OU or O part of the hierarchical name.

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 *Domino.Doc 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 Domino.Doc* provides details on how to configure the servers.

References

http://www.ibm.com/software/lotus/support

http://www.lotus.com/dominodoc

http://www.notes.net

http://www.domino-guru.co.uk

Domino.Doc 3.1 Readme.txt

RAID Without Additional Hardware: Do It Yourself With Windows 2000
http://www6.tomshardware.com/howto/01q3/010906/index.html

RAID brings reliability to mass storage
http://techupdate.zdnet.com/techupdate/stories/main/0,14179,2815047,00.html

IBM Services for Storage Area Networks - An Overview
http://www-1.ibm.com/services/its/us/sanoverview2.html

Designing an IBM Storage Area Network - Redbook
http://www.redbooks.ibm.com/redbooks/SG245758.html

Lamb, John and Peter Lew. 1999. Lotus Notes & Domino 5 Scalable Network Design. New York: McGraw-Hill.

Technote #180280 Instructions for Setting Up Cross-Domain Replication for Domino.Doc

Technote #171301 Domino R5: “Full Text Error (FTG): Exceeded Max Configured Index Size While Indexing....”

Technote #164392 Domino.Doc Prompts for Web Authentication More Than Once When Installed to Subdirectory

© Copyright 1997, 2001 Lotus Development Corporation. © Copyright 2002 IBM Corporation. All rights reserved.
Not for reproduction or other use without express written consent of IBM Corporation.

Part No. CF094NA