Storage Optimization: What's new in Domino 8.0.x and 8.5?

DNUG 5.6.2008 in Bremen
Daniel Nashed, Nash!Com

Nash!Com
Communication Systems

DNUG User Group

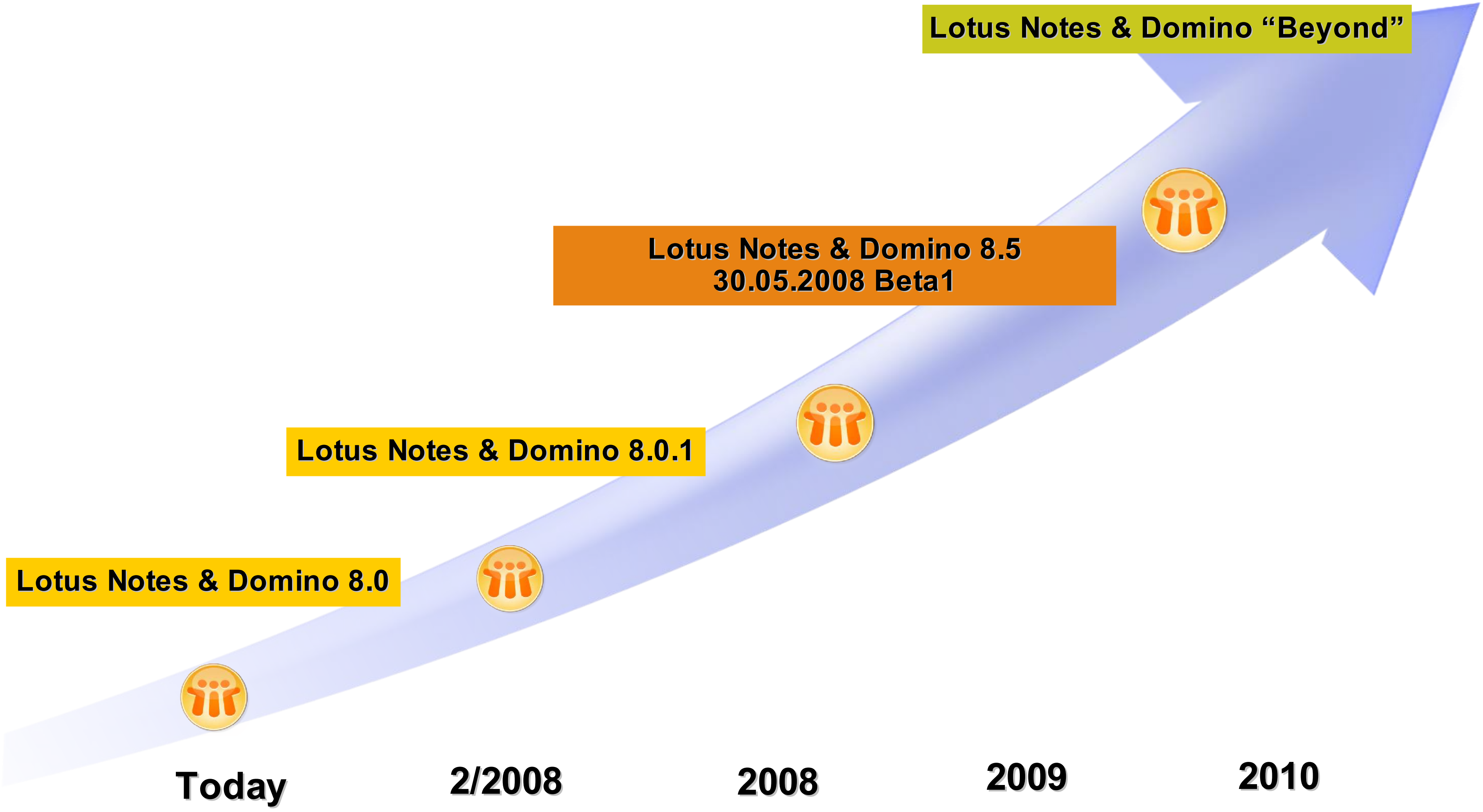
About the presenter

- **Daniel Nashed**
  - Nash!Com - IBM/Lotus Advanced Business Partner/ISV
  - Member of The Penumbra group
    - an international consortium of selected Business Partners pooling their talent and resources
  - focused on Cross-Platform C-API, Domino® Infrastructure, Administration, Integration and Troubleshooting
  - Platform Focus: W32, xLinux, zLinux, AIX® and Solaris®
  - Regular speaker at International Conferences
- **nsh@nashcom.de**
- **http://www.nashcom.de**

Agenda

- **Introduction**
- **Features in Domino 8.0, 8.0.1 and 8.5 Beta1**
  - Design Compression, Data Compression, New Solution for Attachments (DAOS)
  - Other Storage Optimizations
- **First tests**
- **Q & A - ask questions any time**

Roadmap for IBM® Lotus Notes® & Domino®: 2008 and Beyond



Lotus Notes & Domino "Beyond"

Lotus Notes & Domino 8.5
30.05.2008 Beta1

Lotus Notes & Domino 8.0.1

Lotus Notes & Domino 8.0

Today

2/2008

2008

2009

2010

Nash!Com
Communication Systems

DNUG User Group

Before we start

- **Storage Optimization is not the final solution for your storage problems**
  - But it will help a lot stabilizing your environment and gaining performance

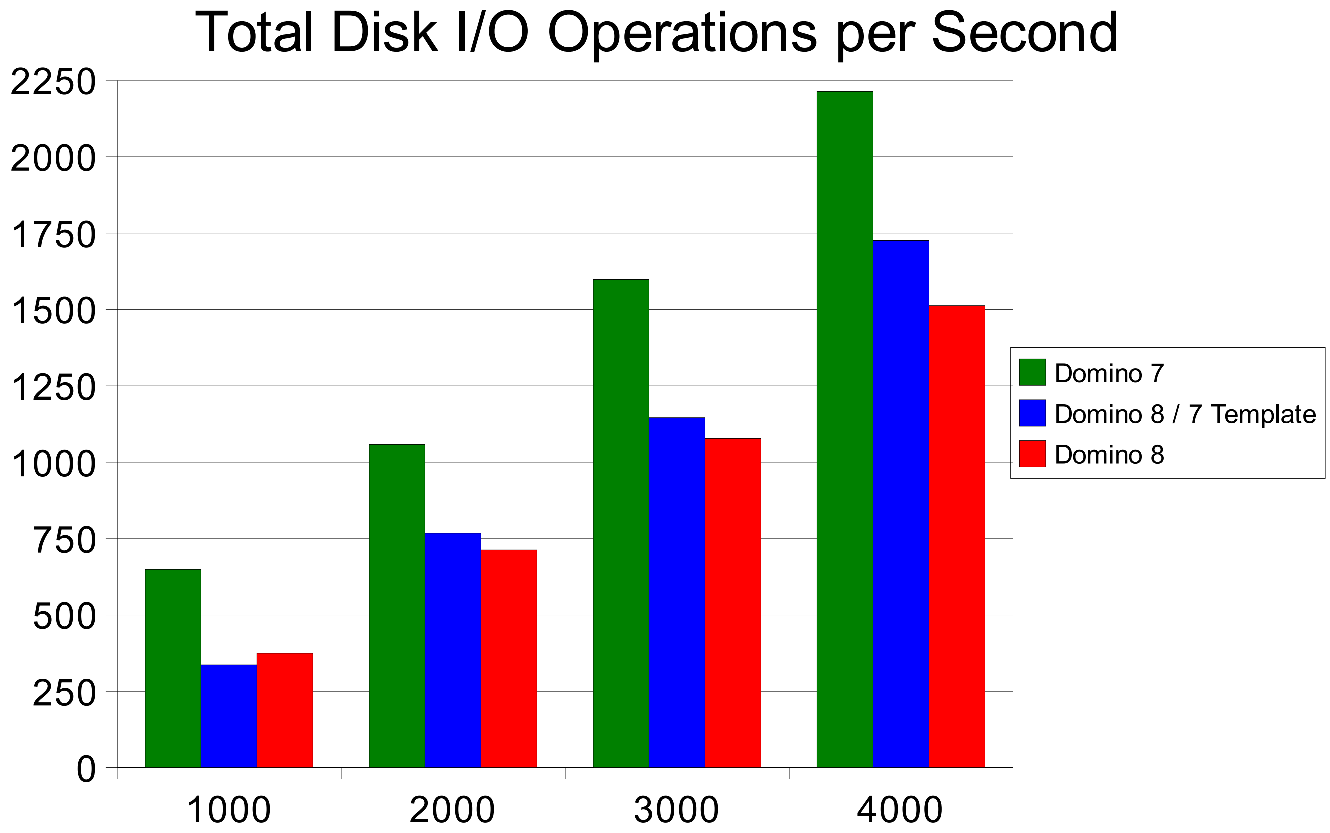
- **Reduce Disk Space Usage**
  - Without Quotas and/or Archiving Mail Size grows exponential
  - Even with current SAN systems disk usage is still a very big pain-point

- **Independent from all storage optimization you have to find an organizational solution**
  - Change the way your users work with their mailfile and provide solutions
  - E.g. Quickr, Discussion Dbs, Archiving, Policies, ..

D8/8.5 Design Goals for Storage Optimization

- **Reduction of Disk I/O**
  - One main aspect is the reduction of I/O requests not just the total amount of I/O
  - Current disk and SAN environments are capable of high I/O transfer rates
    - But the number of I/Os per second (IOPS) is still a limiting factor
- **The goal is to reduce the I/O and keep CPU load on the current level**
  - CPU performance is increasing faster than I/O performance
    - QuadCore vs. current 15K disks
    - SAN can help but higher performance can have exponential cost
  - The bottleneck is still I/O in most environments
    - IOPS are the most important factor

Major I/O Reduction in Domino 8.0



- **Notes:**
  - Windows® 2003 Server results shown. Other platform results are posted on Developerworks
  - Improvements based on Notesbench workload tests and vary by operating system and in customer environment
  - Reductions require new Notes 8 mail template and ODS 48 to be enabled

Lotus Domino 8.0 I/O Improvements

- **Avoid file filling when extending .NSF files**
- **Reduce use of design note access on servers**
  - Used for locating design elements
- **UPDATE task streamlining for unchanged folders**
- **Optimized API for detecting databases changes**
  - Available in D7.0.2 code and used by current BlackBerry releases
  - General Available in Domino 8.0 C-API
- **Streaming Cluster Replicator (SCR)**
  - Details on next slide
- **Notes**
  - Many of those optimizations require use of Lotus Domino 8 ODS 48 databases
  - Has to be enabled thru notes.ini setting (Create_R8_Databases=1)
  - Some optimizations based on use of Notes 8 mail template

D8.0 - Streaming Cluster Replication

- **Default cluster replicator in 8.0**
- **Move from per-database cluster replication to server-wide event model**
- **Replication changes put directly into in-memory queue for processing**
- **Significant source-side CPU reduction - ~10% in benchmarks**
  - But the key factor is the reduced disk I/O
- **Major update latency reduction**
  - Average latency reduced from 269 seconds to 5 seconds in 4,000 user benchmark

D8.0 - Out of Office Services integrated into Mailrouter

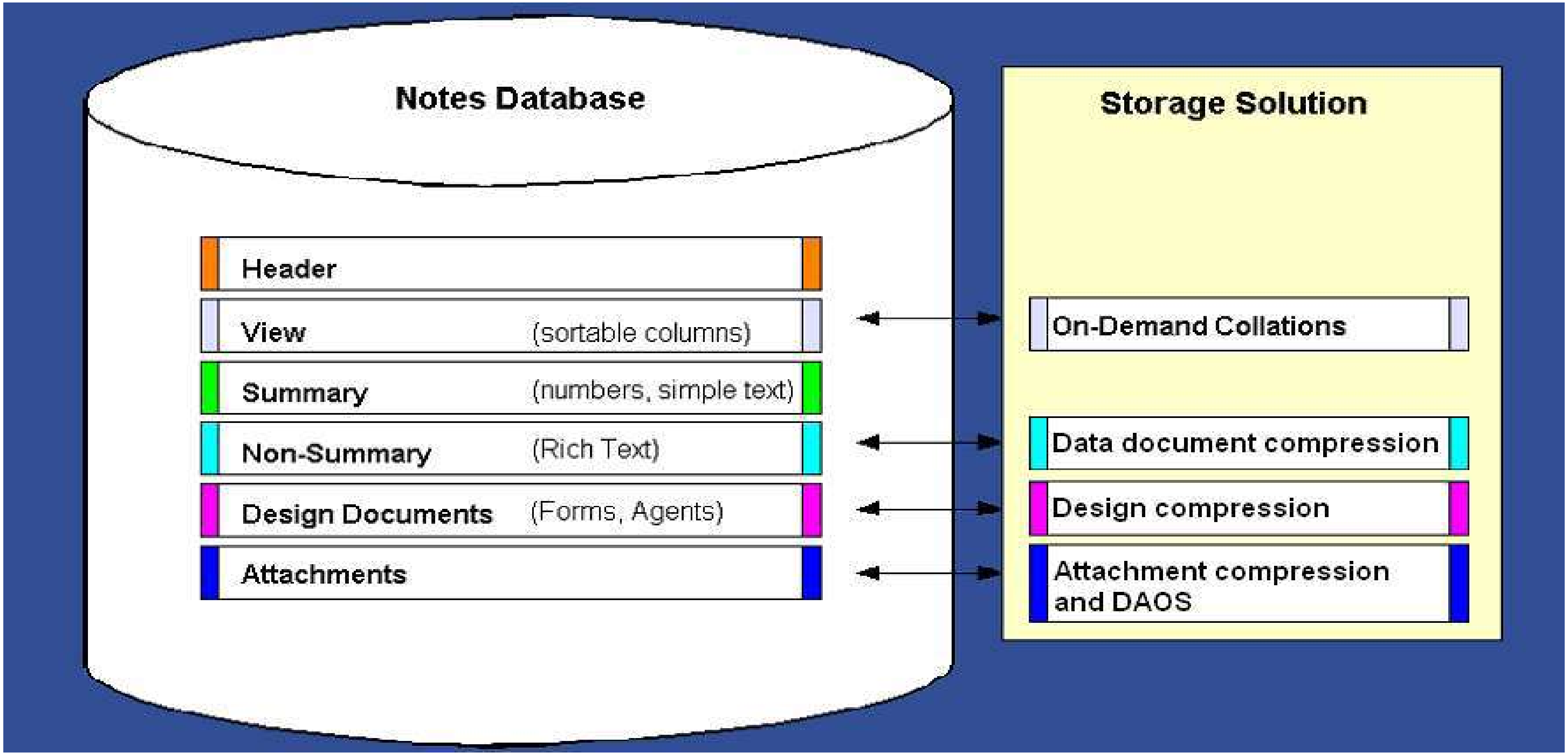
- **Finally! :-)**

- **Out of office runs integrated into the mail-delivery process**
  - Instance response instead of 4 hours delay!
  - Runs on the still open document in memory
  - No need for OOO enabled on mailfiles to read thru unprocessed mail

- **Needs Domino 8.x Server <u>and</u> Notes 8.x Mail-Design**
  - And a couple of settings ...

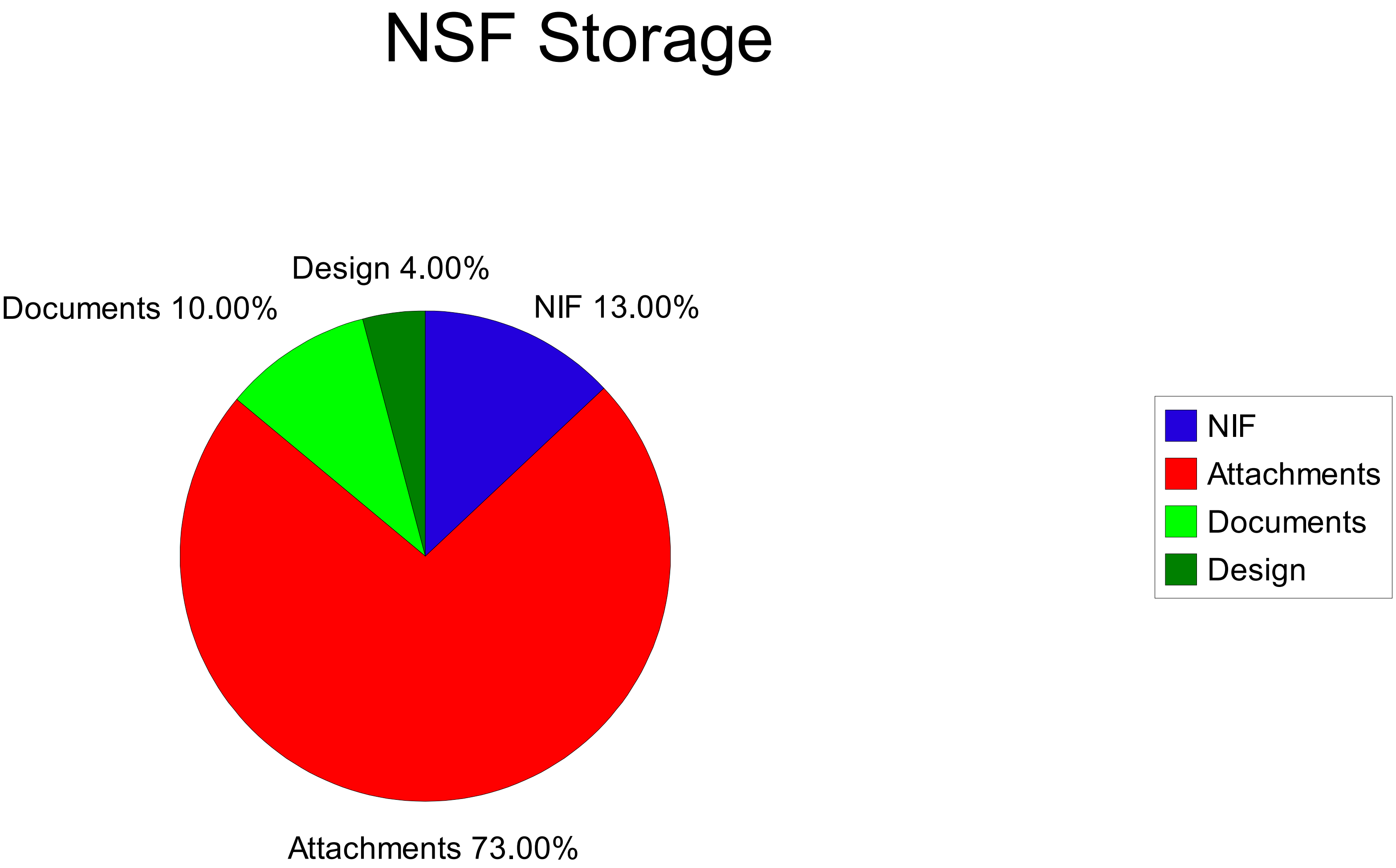
- **Reduces I/O load and also Amgr Load!**

Lotus Domino Storage Reduction Strategy



- **D8.0 - On-Demand Collations, Design Compression**
- **D8.0.1 - Data Document Compression**
- **D8.5 - Attachment Compression**

NSF Storage Content Example Larger DBs (Range between 500 MB and 2 GB)

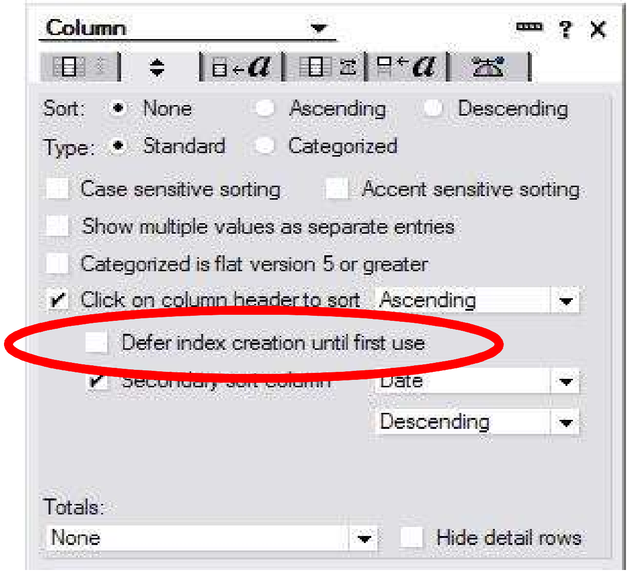


D8.0 - Build-On-First-Use Sort Orders

- **A Notes View can have a primary and multiple "clickable" sort orders (by clicking on the column)**
  - Each of them present a separate "collation" that needs the same amount of disk space and CPU time to be maintained
  - Sorting in both directions doubles the collations!
- **You should avoid having many "collations" per view**
  - BUG: additional collations are not removed and still used for indexing – even you cannot see them in the design any more.
    - Check $collation items in view/folder design
    - Use D7.0.3 Designer or higher to update the design
- **New option in D8.0 to allow additional sort orders to build on demand when first used**
  - When opening a view/folder only the primary sort order is indexed
  - Reduces the time and space for additional sort orders
    - For first build time <u>and</u> incremental updates

D8.0 - Build-On-First-Use Sort Orders

- **You have to specify a new option in view/folder design to benefit from this new setting**
  - In mail8.ntf folders and views in D8.0.1 and D8.5 are already modified out of the box
- **You need to rebuild the views to take benefit**
  - Updall -R or compact -D
- **Huge savings if only the first sort order is used by most of the users**
  - Specially $Inbox and all derived views/folders in mail-file will benefit



Column

Sort: ● None ○ Ascending ○ Descending

Type: ● Standard ○ Categorized

□ Case sensitive sorting □ Accent sensitive sorting

□ Show multiple values as separate entries

□ Categorized is flat version 5 or greater

☑ Click on column header to sort Ascending

□ Defer index creation until first use

☑ Secondary sort column Date

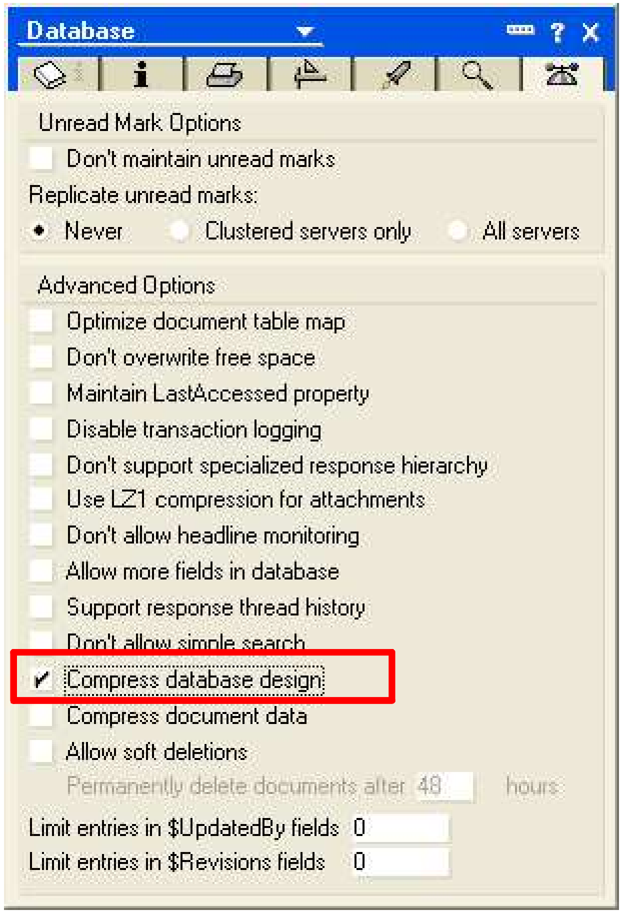
Descending

Totals: None □ Hide detail rows

D8.0 - Design Note Compression

- **New feature of ODS48**
- **Compresses the design of the database**
- **Only the internal storage format in the NSF is changed**
  - No difference at all for any part of the server, client or any C-API code
  - Opening the design notes (e.g. NSFNoteOpen) will automatically uncompress the backend and will pass back a note in normal format
- **For replication and remote accessing the Design the uncompressed note goes over the wire**
  - But you can use Notes network compression on slow lines
    - Take care: Needs to be enabled on Client and Server
- **Reduces the design of a database by 40-50%**

D8.0 - Enabling Design Note Compression



Unread Mark Options

□ Don't maintain unread marks

Replicate unread marks:

● Never ○ Clustered servers only ○ All servers

Advanced Options

□ Optimize document table map

□ Don't overwrite free space

□ Maintain LastAccessed property

□ Disable transaction logging

□ Don't support specialized response hierarchy

□ Use LZ1 compression for attachments

□ Don't allow headline monitoring

□ Allow more fields in database

□ Support response thread history

□ Don't allow simple search

☑ Compress database design

□ Compress document data

□ Allow soft deletions

Permanently delete documents after 48 hours

Limit entries in $UpdatedBy fields 0

Limit entries in $Revisions fields 0

- **Needs ODS 48**
  - notes.ini Create_R8_Databases=1
- **Enable Design Compression**
  - Load compact -n
- **Tip: Compress existing design elements elements in the same run**
  - Load compact -n -C
- **Compression flag has internal representation**
  - DBOPTBIT_xxx
- **And icon note flag that replicates to remote databases**
  - You still need a compact for existing data

Nash!Com
Communication Systems

DNUG User Group

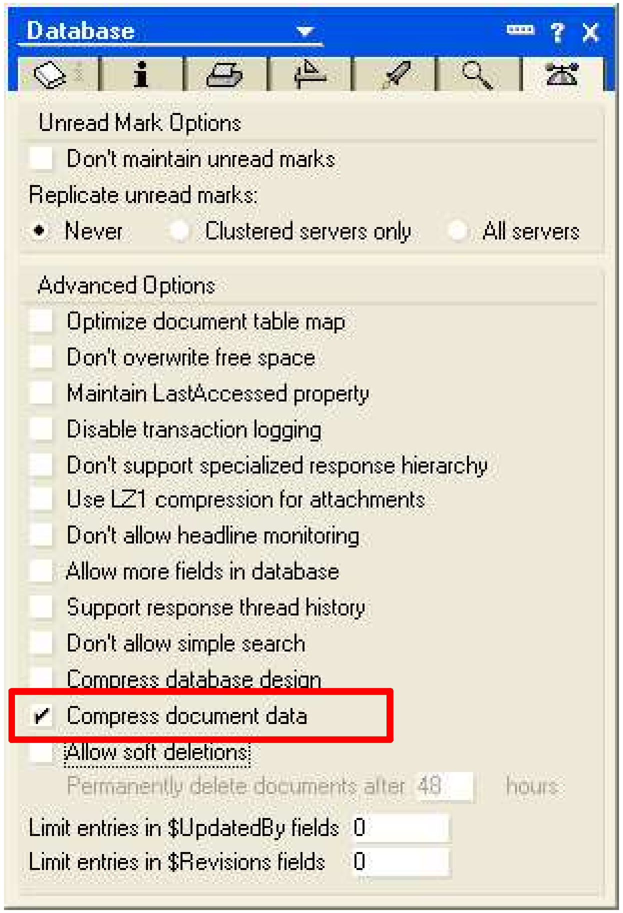
Design Compression Examples

- **Notes 7.0.3 Mail-Template (EN)**
  - Database Size before compression: 18.087.936 Bytes = 17,25 MB
  - Database Size after compression: 8.126.464 Bytes = 7,75 MB
  - **Saving: 55% (9,5 MB)**
- **Notes 8.0.1 Mail-Template (EN)**
  - Database Size before compression: 28.835.840 Bytes = 27,5 MB
  - Database Size after compression: 14.155.776 Bytes = 13,5 MB
  - **Saving = 51% (14 MB)**
- **Reduction for 1000 users = 14GB**
  - But main focus is I/O reduction
- **Use Design Compression instead of Single-Copy Template (SCT)**
  - We have seen a lot of issues with SCT in customer environments
  - Some hints: Enabled agents, refreshing design, moving to a newer release etc…

D8.0.1 - Document Compression

- **New feature of ODS48 in D8.0.1**
- **Similar to design compression but for documents**
  - Summary Data and Richtext (basically all normal items but not the attachments and other objects)
- **Only the internal storage format in the NSF is changed**
  - No difference at all for any part of the server, client or any C-API code
  - Opening the document (e.g. NSFNoteOpen) will automatically uncompress the backend and will pass back a note in normal format
- **For replication and remote accessing also the uncompressed note goes over the wire**
  - But you can use network compression
- **Reduces the size of a document by 40-50%**

D8.0 - Enabling Document Compression



Unread Mark Options

□ Don't maintain unread marks

Replicate unread marks:

● Never ○ Clustered servers only ○ All servers

Advanced Options

□ Optimize document table map
□ Don't overwrite free space
□ Maintain LastAccessed property
□ Disable transaction logging
□ Don't support specialized response hierarchy
□ Use LZ1 compression for attachments
□ Don't allow headline monitoring
□ Allow more fields in database
□ Support response thread history
□ Don't allow simple search
□ Compress database design
☑ Compress document data
□ Allow soft deletions

Permanently delete documents after 48 hours

Limit entries in $UpdatedBy fields 0

Limit entries in $Revisions fields 0

- **Needs ODS 48**
  - notes.ini Create_R8_Databases=1
- **Enable Design Compression**
  - Load compact -v
- **Tip: Compress existing document elements in the same run**
  - Load compact -v -C
- **Compression flag has internal representation**
  - DBOPTBIT_xxx
- **And icon note flag that replicates to remote databases**

D8.0.1 Compression – I/O Reduction

- **Main Focus is I/O Reduction**
  - Kbytes/sec and even more important I/O per second (aka IOPS)
  - CPU utilisation stays the same

- **Data from IBM Lotusphere 2008 Presentaion**

User Txn/min	Response time Seconds	CPU	Disk Ops/sec	Disk Kbytes / sec	
9612	0,044	13,80%	515,7	5679	**No Comp**
9600	0,045	13,80%	398,7	4588	**Comp**
		Reduced by →	**23 %**	**20 %**	

  - I/O utilization with and without Compression
  - Transaction Log Enabled
  - Example from IBM Lotusphere 2008 presentation

Attachment Compression - LZ1

- **Already available since Domino 6**
  - But not widely used because of issues with first versions
- **Should be enabled on mail-databases and server mailboxes**
  - With Domino 8 the SMTP task also leverages LZ1 for incoming mails
- **Compression advantage over Huffman encoding**
- **Only the internal representation of the attachment is optimized**
  - Storage on disk and data transferred over the network
  - For external mails an external ZIP solution can be very useful
- **If you use external ZIP solutions or ZIP files in general you should not have attachments also internally stored compressed**
  - Duplicating compression effort without extra benefit
  - Currently the user decides in attachment dialog if document will be compressed

Recompress existing Attachments

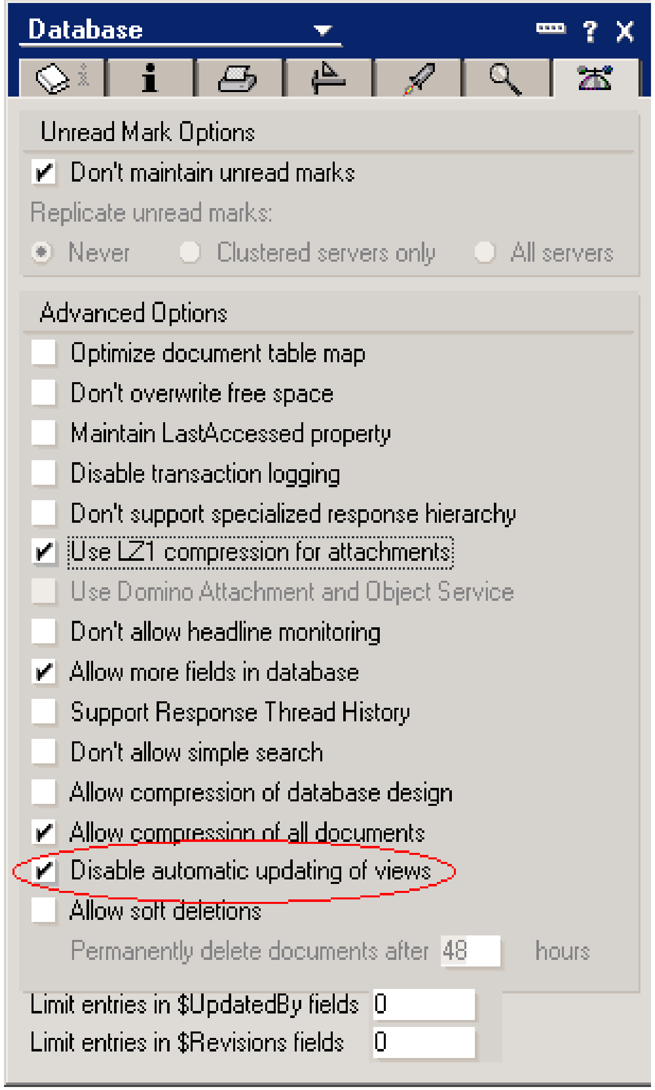
- **In many cases documents are not compressed or compression is inconsistent**
  - E.g. For incoming mail no compressed and Huffman compression
- **Late D7.x and D6.5.x versions have a new compact option to convert Huffman to LZ1**
  - In D8.5 Beta 1 I have seen that also uncompressed files are re-compressed to LZ1
  - But there are still issues with recompression
    - Still working on detailed analysis (Beta1 just shipped end of last week)
- **Compact -ZU -C**
  - Re-compressed the attachments
- **It would be helpful if also the storage encoding would be changed**
  - In some cases the storage is Base64 or Quoted Printable
  - More details about this in the DAOS section of the presentation

What is planned for Domino 8.5?

- **Disclaimer**
  - The following slides give an overview of planned features for storage optimization in D8.5
  - Demos are based on Domino 8.5 Beta1 -- just released end of last week
  - Details are subject to change
  - Some details are not 100% clear yet
    - It's a first beta and there is on-going work

- **You should check the Beta Website to to download and give feedback in the Public Beta Forum**
  - Download
    - https://www.software.ibm.com/webapp/iwm/web/preLogin.do?lang=en_US&source=swg-lnd85
  - Notes/Domino 8.5 Public Beta Feedback forum
    - http://www-10.lotus.com/ldd/nd85forum.nsf

D8.5 - Reduced I/O – Update Task



Unread Mark Options

☑ Don't maintain unread marks

Replicate unread marks:

● Never ○ Clustered servers only ○ All servers

Advanced Options

□ Optimize document table map

□ Don't overwrite free space

□ Maintain LastAccessed property

□ Disable transaction logging

□ Don't support specialized response hierarchy

☑ Use LZ1 compression for attachments

□ Use Domino Attachment and Object Service

□ Don't allow headline monitoring

☑ Allow more fields in database

□ Support Response Thread History

□ Don't allow simple search

□ Allow compression of database design

☑ Allow compression of all documents

☑ Disable automatic updating of views

□ Allow soft deletions

Permanently delete documents after 48 hours

Limit entries in $UpdatedBy fields 0

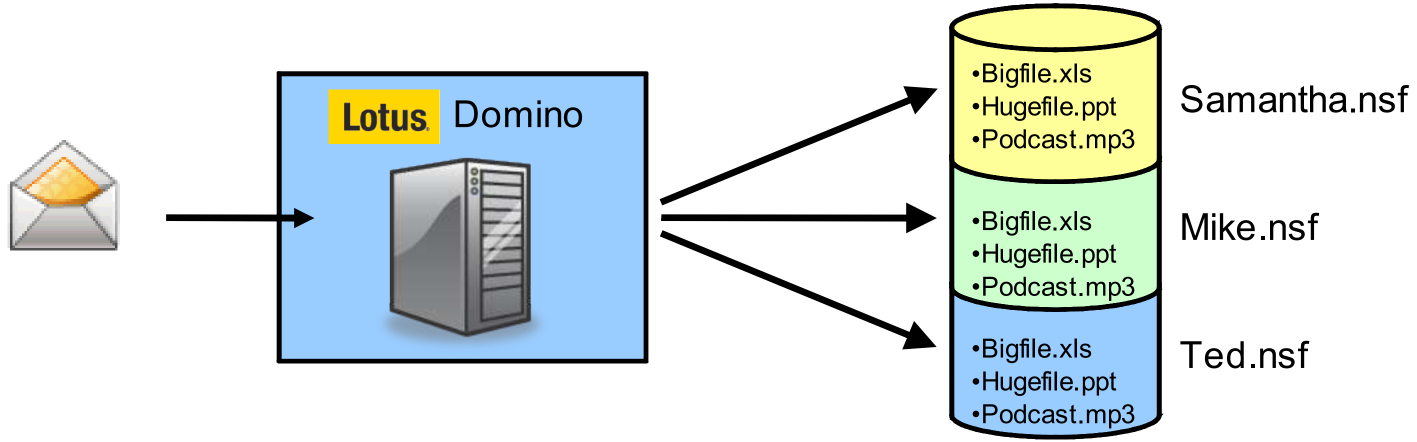
Limit entries in $Revisions fields 0

- **Update Task**
- **Typically very unfavourable cost/ benefit ratio for mail files**
- **Per-database option to opt-out of view refresh**
- **Inheritable from template**

D8.5 - Reduced I/O Mail.Box Optimizations

- **Router Optimization**
  - Handle very large mail throughput
  - Reduce latency of delivery
  - Better parallelization
  - Optimize for transient nature of messages in mail.box
    - Messages generally deleted shortly after arrival
  - Persistent queue mechanism being implemented on top of NSF
- **Leverage transaction log for persistence**
  - Optimize transaction logging operations for Delete operation
  - Reduce transaction log I/O activity
- **Note: Currently translog should be disabled on mail.box!**
  - This recommendation will change with Domino 8.5!
- **Event queue mechanism to avoid searching mail.box for new/ modified messages**

DAOS – Domino Attachment and Object Service



Lotus Domino

•Bigfile.xls
•Hugefile.ppt
•Podcast.mp3

Samantha.nsf

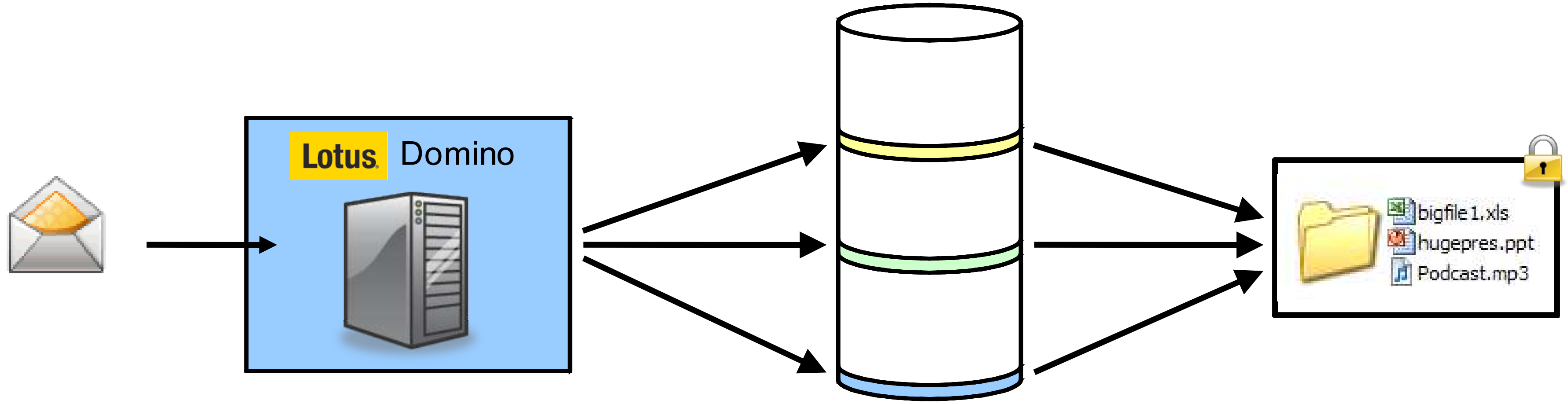
•Bigfile.xls
•Hugefile.ppt
•Podcast.mp3

Mike.nsf

•Bigfile.xls
•Hugefile.ppt
•Podcast.mp3

Ted.nsf

Domino 7 & 8



Lotus Domino

bigfile1.xls
hugepres.ppt
Podcast.mp3

Domino 8.5
with DAOS

Lotus Domino 8.5 – "DAOS"

- **Logically replaces Single-Copy Object Store (SCOS) feature**
  - But has no code in common with SCOS!
  - Fails nicely in case of a problem
  - Works for all types of databases
  - Virtualizes the database backend objects
- **Provides efficient, file-system storage of any type of large object**
  - Automatically removes redundant storage of objects via efficient content comparison.
- **Result:**
  - Databases greatly reduced in size
  - Substantial disk space savings
  - I/O bandwidth savings
  - Huge reductions in backup cost and database maintenance cost
  - Better way to scale and tune disk I/O sub-systems and file-systems

Difference between DAOS and SCOS

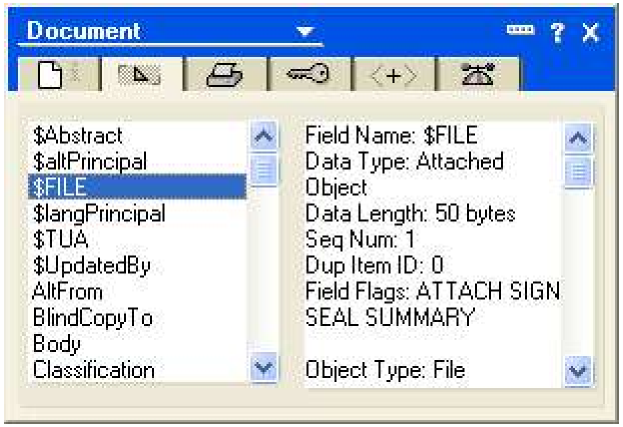
- **SCOS – Single Copy Object Store**
  - Only available for mail
  - Objects are stored in a limited number NSF files
  - If a SCOS backend objects fails you cannot open the document
- **DAOS – Domino Attachment and Object Service**
  - Available for all databases types
  - Only backend object object is transparently moved to the DAOS file store (NLO files) using unique hash keys for matching the file
  - Objects are represented by simple files in the DAOS file-system
  - If DAOS backend object fails you can still open the document
    - Only the missing attachment will cause an error when opening

Lotus Domino 8.5 – DAOS Planned Benefits

- **Simple – Attachments are placed in DAOS based on database setting and size criteria**
  - Reduce disk space depending on degree of content overlap
- **Improve mail performance**
  - Attachments are written once per server
  - Additional users just get a copy of the reference
- **Improve compact performance**
  - Avoids need to move large objects (attachments) during compaction process
  - Only small reference is moved.
- **Reduce file-system fragmentation**
  - Large objects are stored outside the NSF and are normally static
- **Enable large reduction in incremental backup costs**
  - DAOS repository isolates large blocks of data into separate, unchanging files.
- **Resilient – No single point of failure**

How DAOS works

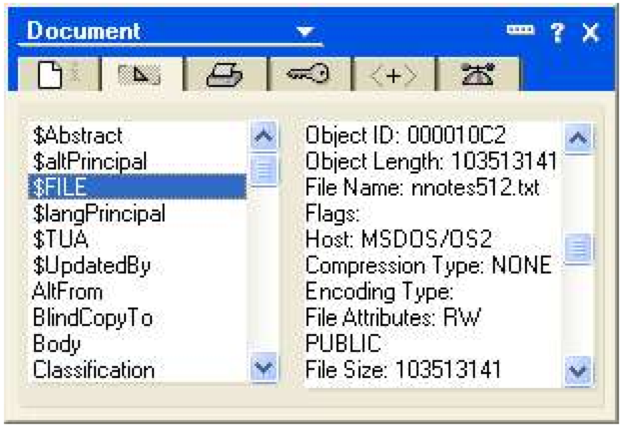
- **Attachments are stored in Objects**
  - $file item contains information about backend object



$Abstract
$altPrincipal
$FILE
$langPrincipal
$TUA
$UpdatedBy
AltFrom
BlindCopyTo
Body
Classification

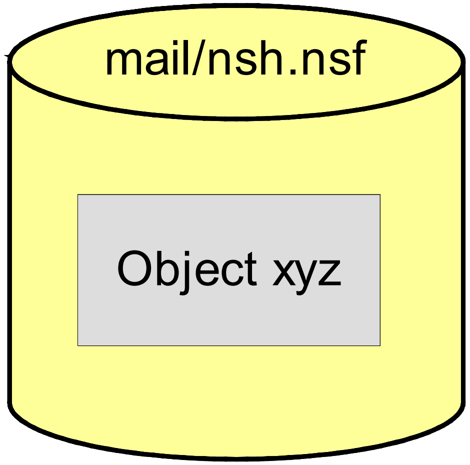
Field Name: $FILE
Data Type: Attached Object
Data Length: 50 bytes
Seq Num: 1
Dup Item ID: 0
Field Flags: ATTACH SIGN SEAL SUMMARY

Object Type: File



$Abstract
$altPrincipal
$FILE
$langPrincipal
$TUA
$UpdatedBy
AltFrom
BlindCopyTo
Body
Classification

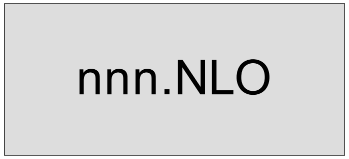
Object ID: 000010C2
Object Length: 103513141
File Name: nnotes512.txt
Flags:
Host: MSDOS/OS2
Compression Type: NONE
Encoding Type:
File Attributes: RW PUBLIC
File Size: 103513141



mail/nsh.nsf

Object xyz

- **With DAOS the object is stored in the file-system**
  - Exactly the same content plus (currently) 115 bytes header
  - File is referenced in the $File item
  - The properties look exactly the same



nnn.NLO

Database Objects

- **Object Definition**
  - Filename
  - Compression Type (None, Huffmann, LZ1)
  - File Attributes and Flags (encodings like Base64, flags for signed objects...)
  - File Size and Object Header (pointer to actual backend object)

```
typedef struct {
  OBJECT_DESCRIPTOR Header; /* object header */
  WORD FileNameLength;      /* length of file name */
  WORD HostType;            /* identifies type of text file delimeters (HOST_) */
  WORD CompressionType;        /* compression technique used (COMPRESS_) */
  WORD FileAttributes;      /* original file attributes (ATTRIB_) */
  WORD Flags;               /* miscellaneous flags (FILEFLAG_, ENCODE_) */
  DWORD FileSize;           /* original file size */
  TIMEDATE FileCreated;        /* original file date/time of creation, 0 if unknown */
  TIMEDATE FileModified;    /* original file date/time of modification */
                            /* Now comes the file name... It is the original */
                            /* RELATIVE file path with no device specifiers */
} FILEOBJECT;
```

Database Objects

- **Actual backend object**
  - Points to an internal object in the database
  - Objects of OBJECT_FILE type are moved to DAOS file-system and internally referenced below the public available C-API
  - This object is moved to the DAOS-Filesystem

```
typedef struct {
  WORD ObjectType;         /* Type of object (OBJECT_xxx) */
  DWORD RRV;             /* Object ID of the object in THIS FILE */
} OBJECT_DESCRIPTOR;

#define OBJECT_FILE       0  /* File Attachment */
#define OBJECT_FILTER_LEFTTODO      3  /* IDTable of "done" docs attached to filter */
#define OBJECT_ASSIST_RUNDATA     8  /* Assistant run data object */
#define OBJECT_UNKNOWN 0xffff
```

Lotus Domino 8.5 – DAOS FAQ Answers

- **DAOS is local to the Domino server – not cross server**
  - Streaming operations (replication, storing documents) is still a 32K blocked streaming operation
  - Temporary NLO files are generated (with 15 zeros in the name) that result either into a new permanent file or will be deleted if the NLO file already exists (ref count update)
- **DAOS requires transaction logging to be enabled on the Domino server and for participating database**
- **DAOS requires a new database ODS 50**
  - Enabled via notes.ini Create_R85_Databases=1 and Copy-Style Compact (-C)
- **DAOS is API transparent**
- **DAOS objects count against quotas and are reported in the file size**
  - Quota Structure in database is maintained internally

DAOS Components

- **Server Doc settings**

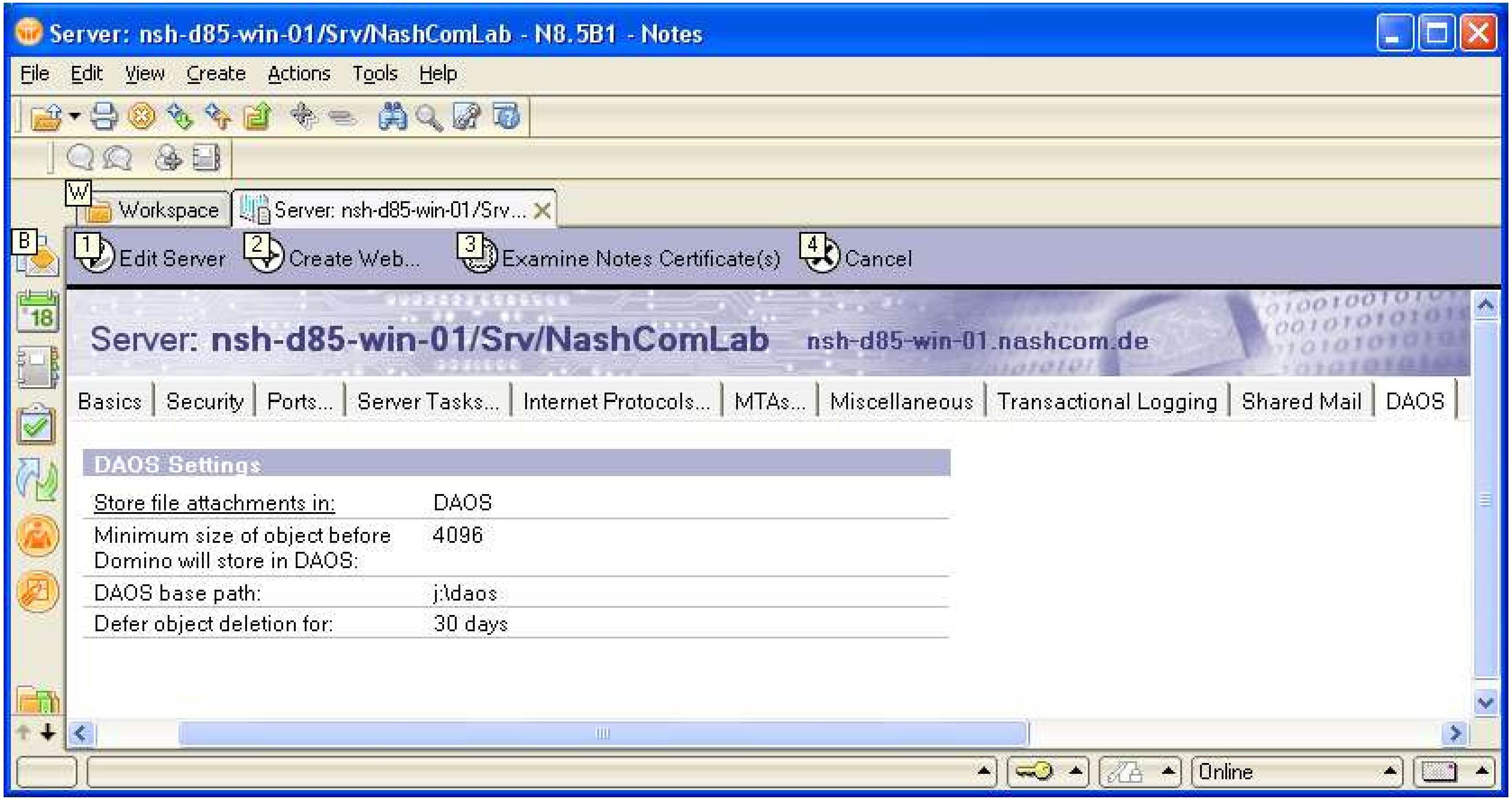
- **DAOSCAT.NSF**
  - Internal reference count database
  - Has no design nor documents and is for internal use only
  - There is currently no access planned for this data

- **DAOS.CFG**
  - DAOS configuration stored in XML format (located in data directory)

- **DAOS file-system**
  - Contains the NLO files in folders named 0001 thru 9999 containing up a couple of thousand files each

How to enable DAOS?

- **Enable DAOS in Server Doc**


Server: nsh-d85-win-01/Srv/NashComLab - N8.5B1 - Notes
File Edit View Create Actions Tools Help
Workspace
Server: nsh-d85-win-01/Srv...
1 Edit Server
2 Create Web...
3 Examine Notes Certificate(s)
4 Cancel
Server: nsh-d85-win-01/Srv/NashComLab nsh-d85-win-01.nashcom.de
Basics | Security | Ports... | Server Tasks... | Internet Protocols... | MTAs... | Miscellaneous | Transactional Logging | Shared Mail | DAOS
Online

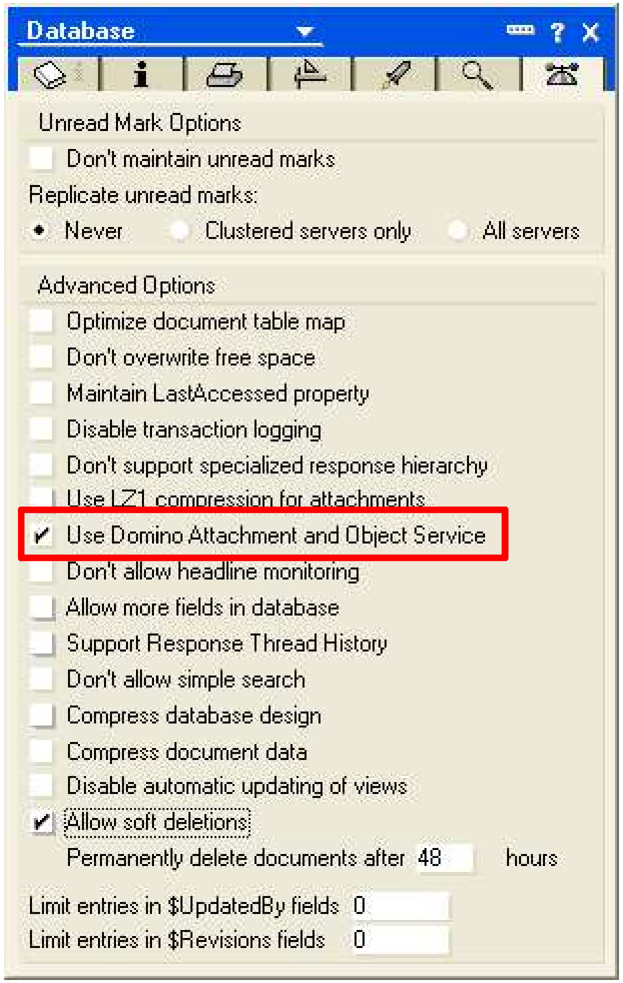
DAOS Settings

Store file attachments in:	DAOS
Minimum size of object before Domino will store in DAOS:	4096
DAOS base path:	j:\daos
Defer object deletion for:	30 days

DAOS Detailed Settings

- **Specify a separate RAID disk or separate SAN LUN for your DAOS store**
  - Tuning for this file-system is still not determined
  - Access pattern probably more sequential large file I/O in 32K shunks
  - Probably best to use RAID5 striped 128K
- **Think about at what size threshold you want to store files in DAOS**
  - Small threshold could lead to a very very high number of files in the file-system
  - Default is 4K. Maybe a higher threshold makes sense in your environment
    - E.g. 128KB, 512KB, 1MB?
- **You should set the Deferred Object deletion higher than your backup retention time**
  - This makes backup restore procedures a lot easier

D8.5 - Enabling DAOS



Unread Mark Options

□ Don't maintain unread marks

Replicate unread marks:

● Never ○ Clustered servers only ○ All servers

Advanced Options

□ Optimize document table map

□ Don't overwrite free space

□ Maintain LastAccessed property

□ Disable transaction logging

□ Don't support specialized response hierarchy

□ Use LZ1 compression for attachments

☑ Use Domino Attachment and Object Service

□ Don't allow headline monitoring

□ Allow more fields in database

□ Support Response Thread History

□ Don't allow simple search

□ Compress database design

□ Compress document data

□ Disable automatic updating of views

☑ Allow soft deletions

Permanently delete documents after 48 hours

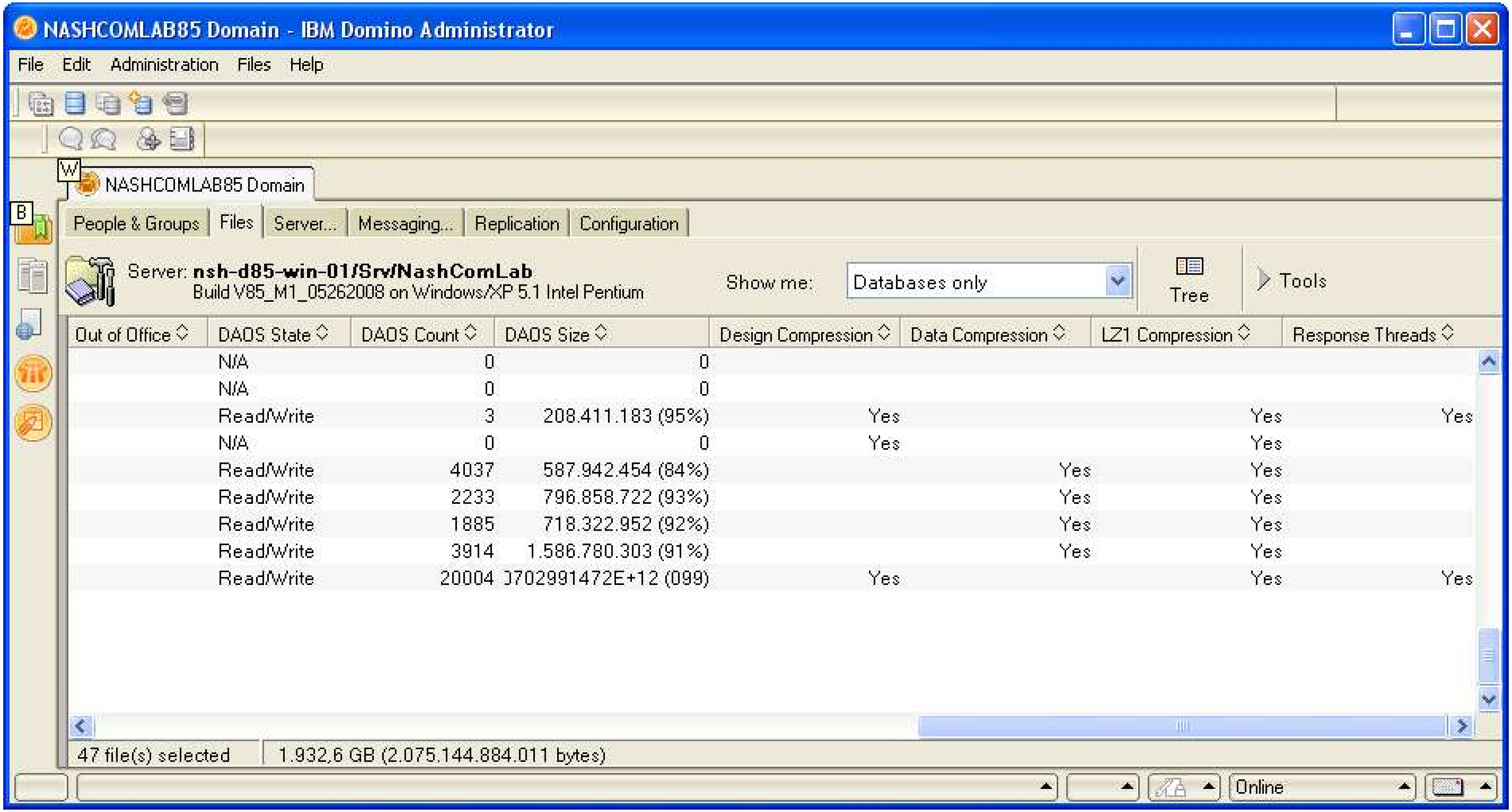
Limit entries in $UpdatedBy fields 0

Limit entries in $Revisions fields 0

- **Needs ODS 50**
  - notes.ini Create_R85_Databases=1
- **Enable DAOS**
  - Load compact -daos on
- **Tip: Move existing attachments to DAOS in the same run**
  - Load compact -daos on -C
- **DAOS setting has internal representation**
  - DBOPTBIT_xxx
- **And icon note flag that replicates to remote databases**

N8.5- New Columns in Admin Client

- **New columns in Files Tab Admin Client**



NASHCOMLAB85 Domain - IBM Domino Administrator

Server: **nsh-d85-win-01/Srv/NashComLab**
Build V85_M1_05262008 on Windows/XP 5.1 Intel Pentium

Show me: Databases only

Out of Office	DAOS State	DAOS Count	DAOS Size	Design Compression	Data Compression	LZ1 Compression	Response Threads
	N/A	0	0				
	N/A	0	0				
	Read/Write	3	208.411.183 (95%)	Yes		Yes	Yes
	N/A	0	0	Yes		Yes	
	Read/Write	4037	587.942.454 (84%)		Yes	Yes	
	Read/Write	2233	796.858.722 (93%)		Yes	Yes	
	Read/Write	1885	718.322.952 (92%)		Yes	Yes	
	Read/Write	3914	1.586.780.303 (91%)		Yes	Yes	
	Read/Write	20004	J702991472E+12 (099)	Yes		Yes	Yes

47 file(s) selected | 1.932,6 GB (2.075.144.884.011 bytes)

Online

Storage Gains

- **Design Compression**
  - 40-50% of document storage
- **Data Compression**
  - 40-50% of the database design
- **Re-Compression of attachments via Compact -ZU**
  - Estimate: Depending on your data could be a potential reduction by 10-30%
    - Will need compact -ZU work for all types of compression types and maybe encodings
- **Attachments via DAOS**
  - Highly depends on your data
  - Estimate 20-30% with re-compression
  - Reducing the total storage on disk is just one aspect of DAOS
    - More important is the reduction of the NSF storage
    - About 100% of attachment storage is moved from NSF to the DAOS file-system

Lotus Domino 8.5
Further Reduction in I/O Rates

- **Goal – Further 50% reduction in I/O rates for operations and bytes moved over 8.0**

- **Further improvements made or in progress so far:**
  - Document compression
  - Per-database control for Update task
  - Very large reduction in cost of Schedule and Design tasks
  - Mail.Box-specific optimizations to minimize or avoid disk writes

- **Large set of transaction log I/O rate reductions**

- **Better grouping of I/O on smaller set of database pages**

Domino 8 Tips

- **For supported configurations see**
  - TN #1264690 Supported configurations for Notes and Domino 8.0
  - Contains detailed information with Domino, Notes and Template release is supported in combination

- **Domino 8 uses a new licence check routine (LUM)**
  - Licence file ships with Domino but in some cases with multiple releases it cannot be found.
    - Should be located in data directory (Domino8.lic)
    - Checks Registry for location of file

- **Disable transaction log on mail.box in D7.x and D8.x**
  - You need a offline compact -t when the database is not in use!
  - Setting the property is not sufficient!
  - notes.ini: MailBoxDisableTXNLogging
    - New mail.box databases will be created with translog disabled

Q&A

- **Questions?**
  - Now, find me later at the conference or contact me offline

- **Contact**
  - nsh@nashcom.de
  - http://www.nashcom.de
  - +49 172 2141912