Lotusphere 2010

Lotus knows.

Smarter software for a Smarter Planet.

# BP204 “CSI Domino” -- Diagnostic Collection and Analysis

**Daniel Nashed** | Nash!Com



lotusknows.com

# 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, Lotus Domino® Infrastructure, Administration, Integration and Troubleshooting
  - Platform Focus: Win32/Win64, Linux®, AIX® and Solaris®
  - Regular speaker at International Conferences
  - DNUG Enthusiast IBM Lotus® Domino Infrastructure

- nsh@nashcom.de
- http://www.nashcom.de

# Agenda

- Introduction – What is „Serviceability“
- Diagnostic Collection / Monitoring
- Crash Analysis, Hang Analysis, Trapleaking
- Performance Analysis
- Q&A

# What is Serviceability?

- RAS = Reliability Availability Serviceability
  - RAS is the effort to improve the Lotus Domino product suite so that:
    - Client/Server doesn't crash or hang as often (Reliability)
    - Client/Server performs well, Server is available to clients (Availability)
    - The ability to quickly pin-point and fix problems (Serviceability)
- Ongoing effort in each incremental release
  - Some features are even back-ported from Lotus Domino 8 to Domino 6&7
- It's not just about NSD & Memcheck but all parts of Domino
  - Logging, Debug Options, etc
- Great help for Admins, Developers and Troubleshooters
  - Too many features and options for a single session ... but we will try ...

# Diagnostic Features in Lotus Domino

- Directory \IBM_TECHNICAL_SUPPORT
  - Single place of log files collection for all the various trace and debug options
- Automatic Data Collection / Configuration Collector
  - Server and Client mail self-acting, configuration snap-shot
- Domino Domain Monitoring (DDM)
  - Comprehensive Server Monitoring
- Dynamic Console Log
  - Log file containing all log and DEBUG information
- NSD
  - Notes System Diagnostics, Memcheck
  - Contains a lot of details about a crash, open databases, Domino and system configuration
- Fault Recovery
  - Generates NSD files and restarts servers automatically
- Memory Dumps, Trapleaking, Semaphore Debugging, ...

# Fault Recovery

- Domino Server detects crash and restarts automatically
  - Panic routine calls fault recovery code
- Enabled in Server document
  - Run NSD To Collect Diagnostic Information: Enabled
  - Automatically Restart Server After Fault/Crash: Enabled
  - Mail Fault Notification to: LocalDomainAdmins
- How does fault recovery work
  - Run NSD + memcheck if configured
  - Cleans up resources
  - Restarts Server
- Very important to enable fault recovery and NSD collection!
  - Server is back online quickly and you have diagnostic data from every crash!

# Automated Diagnostic Collection (ADC)

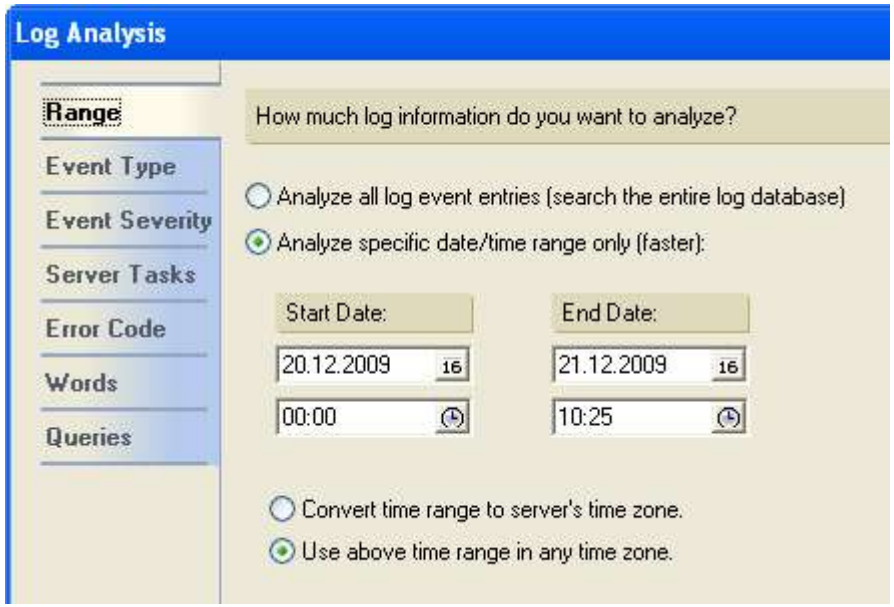
- Enables you to set up a mail-in database to collect the diagnostic information generated from the ND Client/Server crashes into central repositories.
  - Senddiag servertask runs on startup to collect information like NSDs
- Server Configuration Doc / Diagnostics Tab
  - Fault-Recovery Database (lndfr.nsf) as Mail-in Database
    - Size for diagnostic data, retention days, ...
    - Filter pattern to add to data collection (file-patterns!)
    - Enable FaultAnalyzer (since D7) for Fault Database
- Fault Recovery Database and FaultAnalyzer are typically allocated on admin server
  - „FaultAnalyzer“ Servertask
  - Used to collect annotate, categorize NSDs
  - Similar call-stacks, Same Domino releases, Client or Server
  - Have separate databases for clients / servers

# Domino Domain Monitoring (DDM)

- Comprehensive Monitoring
  - ddm.nsf contains focused monitoring results
  - Detailed error messages including names of resources
  - Suggestions for problem solution including actions!
- Based on the foundation build by event monitoring
  - Event categorization and severity defined in events4.nsf
- Additional build in probes into the code
  - Replication (detailed reporting for failing replication)
  - Agent Manager (long running agents, high memory/CPU usage, ...)
- You can also leverage statistics and platform statistics
- Helps to get focused information from all Domino servers in a central location
  - Build-in workflow to assign issues and keep track what has been already solved
- DDM would be a whole presentation of its own

# Analysis Tools

- Domino Admin Client contains analysis Tools
  - Located in Server/Analysis Tab
    - Cluster Analysis
    - Log Analysis




- You should regularly analyze server logs in addition to daily DDM checks

- Activity logging can also help for troubleshooting
  - Needs to be enabled in Server Config Document

# HTTP Diagnostic

- domlog.nsf
  - Response time in each request
  - Allows to filter requests (request-types etc, configured in the server doc)
- Tell http dump config
  - Writes HTTP config to IBM_TECHNICAL_SUPPORT/httpcfg.txt
- tell http debug session on|off
  - Session debug logs
- tell http debug thread on|off
  - Thread debug logs.
- Tell http debug postdata on|off
  - Post data to debug logs.
- Tell http debug responsedata on|off
  - Logging of response content to
- Tell http debug outputio on|off
  - logging of network output tracing

# NSD

- Only invoked automatically when fault recovery is enabled on server
  - Can be started manually if server has already crashed but not yet recycled
  - Can also be used to terminating a hanging server ( nsd -kill )
    - e.g. remove shared memory, semaphores and other resources...

- Can be used on running servers for troubleshooting and server hang diagnostics
  - Does not crash a running server
    - If you have the right OS patchlevels!!!
    - Caution: Windows2003 Server SP1 or higher required for detaching from running processes!

- Invoked at server startup to take a snapshot of the current environment
  - Sysinfo NSD contains all details of your configuration

# Major Sections of an NSD in Detail

- Header: Version and System
- Process Table  / Active Users
- Call-Stacks of running Processes
- MEMCHECK: - Notes / Domino Memory Analyzer
- Shared memory handles and blocks
- Open Databases, Open Documents
- Performance Data
- notes.ini
- User OS-level Environment

# Major Sections of an NSD in Detail

- Executable & Library Files
- Data Directory Full Listing
- Local Disks
- Memory Usage
- Network Stats
- Active Connections, Ethernet Stats, Active Routes, Protocol Stats
- Core File (in some cases)
- Sometimes NSD invokes a memory dump
- OS-Specific information
  - Installed software, Configuration, etc

# NSD Update Strategy

- NSD & Memcheck are updated in each release
- Changes are incorporated into new releases and are available for older Domino releases thru special hotfix installer
- NSD/Memcheck Code is build independent from Domino release
  - See TN #1233676 - NSD Fix List and NSD Update Strategy
  - See TN #4013182 - Updated NSD for Domino releases
    - Contains FTP download links
- You should keep NSD up to date!
- Too many details to list on a single slide ...
  - Improvements and fixes in each dot release or fixpack

# Run NSD as a Service

- New Feature in Lotus Notes/Domino 8 allows NSD to run as a service
  - Avoids issues with OS level users not having proper access to subdirectories or ability to attach to system processes
  - One instance of NSD will run in background continuously as a service
  - When a crash occurs, or NSD is run manually, dynamically created instance of NSD will proxy the request to start NSD Service
  - Specially important also on Citrix environments or other clients with limited OS level access

- Details in Domino 8 Admin Help and NSD HTML help
  - nsd
    - -svcinst | –svcuninst
    - -svcstart | -svcstop
    - -svclog | -svcreport
    - If NSD service is started it is used automatically

# Why Server Freeze and Server Panic?

- Lotus Domino uses shared memory to allocate global resources to share between tasks and Domino core for different sub-systems
  - NIF, NSF, ... e.g. views are stored in memory ...
  - Currupt Memory-Handle or other Handles can have impact on other running tasks and result in corrupted databases

- Lotus Domino "halts" the Server or Client with a PANIC or Freeze to avoid further damage
  - Freezing all tasks / threads
  - Diagnostics and Recycle Routines are called to restart

# What can cause server crashes?

- Design Elements / LotusScript/Java™
- Third Party code
- Corrupt data
  - Corrupt documents, etc ...
- Memory Management issues
  - Overwrites, handle locking, memory leaks)
- Insufficient Memory
  - Often caused by „Memory Leaks“

# First Steps Analyzing a Crash

- Find the crashing thread
  - **"Fatal"** is the most common indication of the crashing task
  - If you don't find fatal, look for "Panic", "Access Violation" or "Segmentation Fault", "Signal" messages on Unix/Linux
  - Tip: Last lines on console.log is helpful in most of the cases
    - Included in current versions of NSD as a separate section

- Analyze the calls in the call-stack
  - It is helpful to know about the C-API toolkit (SDK) to understand function names and parameters involved
  - Not all function calls are exposed
    - But the SDK (C-API Toolkit) gives you a good idea what to look for

# Reproducible Call-Stack/Bug?

- Best case scenario: Reproducible call-stack on independent machines which does not occur on boxes with other releases

- But we are not always that lucky ...
  - If the call stack is similar at the end of the stack it could be a low-level API problem
  - If the call stack is similar at the higher level of the stack always in the same Servertask it could be the Servertask
  - If you see EM_BEFORE, EM_AFTER it might be an Extension-Manager problem
  - If it is always the same database it might be a data problem

- Find open databases
  - You can find open databases by matching the physical/virtual thread-ID with the memcheck section

# More Information about Open Files/Documents

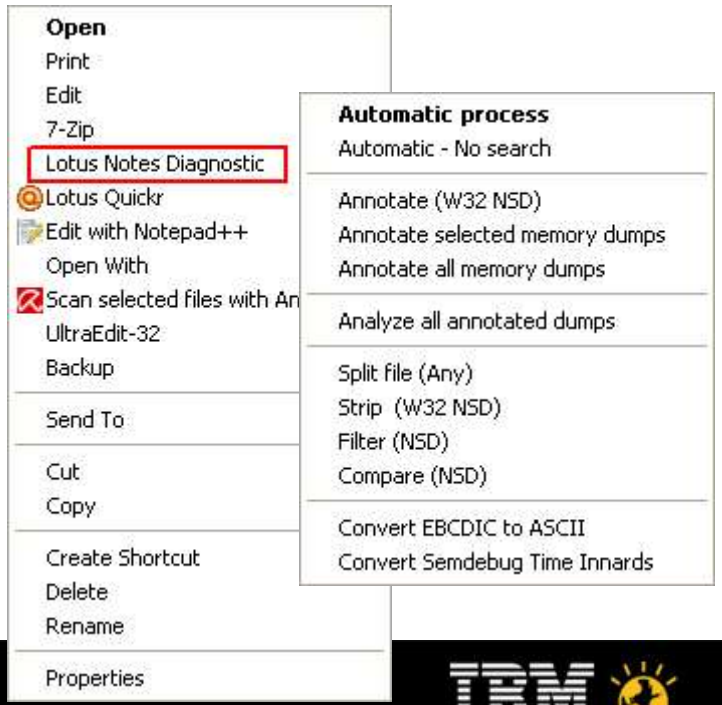
- Check "Resource Usage Summary" section
  - Lists all open DBs for every thread .. with handles and users
- Check "Open Database Table" section
  - Other open databases in the same task at the same time
- Check "NSF DB-Cache" section
  - Databases open in Cache
- Check "Open Documents" section
  - Open Documents with matching database handles

# Lotus Notes Diagnostics

- Tool to annotate NSDs, semdebug files, memory dumps etc
- Current Version 2.8 downloadable from IBM
  - http://www.ibm.com/support/docview.wss?rs=899&uid=swg24019151
- Can be used to annotate crash NSDs

- Ships with notes database, plugs into Explorer

- Very helpful tool
  - Helps you to find crashing call-stack and categorizes the various NSD sections
  - Also matches the data section of the thread in memcheck
  - But you still have to know much about the background to interpret the results



# Abnormal Process Termination -- Also a Crash

- Server task simply disappears from the OS process list with no errors produced (very rare)
  - Domino Server console indicates the task is still running
  - Task cannot be shutdown cleanly from console
  - Unix/Linux: ChildDied Signal on also kicks in fault-recovery
  - If process monitor notices a sub-process not cleanly terminated fault-recovery is also invoked
- Must be treated as a crash
  - Background: Could cause major problems like semaphore hangs, resources that are not cleaned up etc...
- Troubleshooting:
  - Start/stop task debugging: **debug_initterm=1**
    - Logs start/stop of tasks

# Next Steps

- Customer can only fix data problems, check/add server resources (e.g. memory) or install later versions
- IBM Support can look into SPR database and find matching call-stacks
  - Support needs all information available in IBM_TECHNICAL_SUPPORT directory
    - please ZIP files!
  - Every new version of Domino provides more diagnostic information (NSD, ADC, ...)
- Development or 3rd party software vendor can identify new problems and look into their source code
  - Take care: NSD also contains some sensitive information about your system and users.
  - Check the NSD before sending it to external people
  - Add-On Applications on Windows need to have own Domino formatted “SYM” files for call-stack annotation in NSD

# Server Hang Symptoms

- Server (or specific task) is still running, but client receives error messages "Server not Responding"
  - No error is produced on the console but an error may be written to log.nsf
- Console does not accept keyboard commands
- Servertask will not shutdown cleanly
- User report that other Domino server tasks have slowed down
- No NSD is generated and no Fault Recovery

# What can cause hangs?

- LotusScript/Java™
  - Looping logic in code
- Semaphore issues
  - Deadlocks, low level looping
- Permanent unavailability of a particular resource
- Third Party code (FT file-filters)
- General: OS-level calls which do not return to the calling Domino code
- Network issues (DNSLookup, port problems)

# How to troubleshoot Server Hangs?

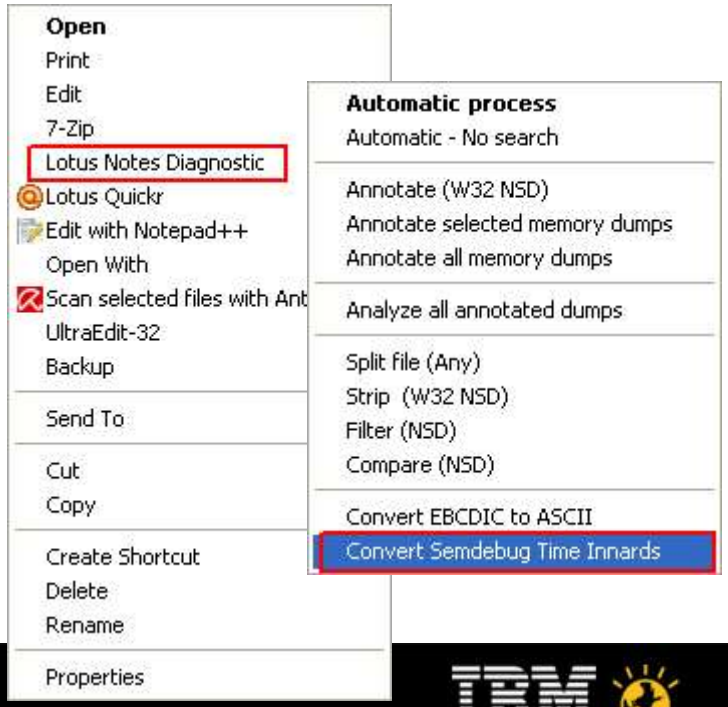
- Check call-stacks for specific calls
  - e.g. a large number of Semaphore Calls, SpinLock Calls
- Use Semaphore Debugging
  - DEBUG_SHOW_TIMEOUT=1
  - DEBUG_CAPTURE_TIMEOUT=10
  - DEBUG_THREADID=1
  - Optional: DEBUG_SEM_TIMEOUT=X
    - (in milliseconds, default 30000)
- Run 3 nsd -nomemcheck in short sequence
  - plus one full NSD
- „Show stat Sem.Timeouts“ to check semaphores
  - Only works with semaphore debugging enabled and only gives you a quick summary

# Analyzing Semaphore logs

- semdebug.txt in IBM_TECHNICAL_SUPPORT
  - contains semaphores locked for more than 30 seconds
  - Information about process/thread, semaphore, time, ...
  - Also contains information who is currently holding the semaphore
  - But just the process/thread.id – You have to annotate on your own via NSD
  - What is always important is the call-stack of the process requesting and olding the semaphoere
  - Can only be done thru NSD

- Example:
  - ti="0025CA9C-C1257353" sq="00004CE8"
    THREAD [28208:00241-169659312]
    WAITING FOR SEM 0x0931 Task sync semaphore
    (@0F7711A4) (OWNER=28208:158743472) FOR 5000 ms

- ti is the internal repesentation of the timedate
  - You can use LND to annotate the ti values



# Memory Consumption / Memory Leaks

- Domino has only a certain amount of addressable memory for
  - Shared Memory
  - Local Process memory
- The limit depends on the platform
- Combination of shared memory + local process memory is the limiting factor
  - For 32bit the total limit is 4GB at most
  - The larger part of memory used is shared memory
- You can run into peek memory situations
- Or run into memory leaks
  - Memory not released when the application does not need it
    - Certain Memory type (block) grows beyond reasonable numbers

# Memory Dumps

- Domino uses an own Memory Management Layer
  - Different Memory types
  - Pooled memory (DPOOLS)
  - Direct memory allocations
- Memory is managed by Domino
  - Allocated Pool memory will be freed to Domino Memory Manager not Operating system
  - Memory Allocation can be tracked and troubleshooted
- Local and Shared Memory
  - Shared Memory for all Servertasks
  - Local Process Memory per tasks
- Different memory block types for each part of the server
  - Each block type (BLK_xxx) can be tracked separately

# Memory Dumps

- You can dump memory
  - Run server -m
  - Or „show memory dump“
- Memory Dump contains
  - Shared/Local Process memory
  - Block Codes
  - Size
- Can be used to determine memory bottlenecks and leaks
- Memcheck output also provides details about memory
  - Check the „**Top 10**“ Sections in NSD as a quick info about memory allocations

# Memcheck Top 10 Memory Section

- There is a TOP 10 Memory section for shared memory
- And a TOP 10 local memory section per servertask

```
 <@@ ------ Notes Memory -> Usage Summary -> Top 10 Memory Block Usage -> Memhandles By Size
:: (Shared) (Time 15:05:03) ------ @@>

   Type  TotalSize      Count    Typename
 ---------------------------------------------------------
 0x82cd  535330816        136  BLK_UBMBUFFER
 0x8472   15733654          1  BLK_DTRACE
 0x82cc    9922560        136  BLK_UBMBCB
 0x8252    5242880          5  BLK_NSF_POOL
 0x834a    3670464          4  BLK_GB_CACHE
 0x8a05    3300000          1  BLK_NET_SESSION_TABLE
 0x83e4    2097152          2  BLK_LKMGR_POOL
 0x8311    2097152          2  BLK_NIF_POOL
 0x93ad    1260162        138  BLK_VA_UNKDESC
 0x826d    1048576          1  BLK_NSF_DIRMANPOOL
```

# Memory Trap Leak Debugging

- Once you figured out about a problematic Memory Block Type you can enable Trap Leak Debugging
  - Debug_Trapleaks=**0x3A45**
  - Debug_Trapleaks_ShowStack=1
  - DEBUG_SHOWLEAKS=1
  - DEBUG_DUMP_FULL_HANDLE_TABLE=1
  - DEBUG_DUMP_BLOCKCODES=1
  - **DEBUG_TRAPLEAKS_NEW=1**
    - Summarzies call-stacks
- Checks Memory allocations and dumps call-stacks
  - when task is shutdown (local memory)
  - when server is shutdown (shared memory)

# Backup Memory Limitations

- It's not always a memory leak
- Shared Memory is limited to 2 – 3 GB depending on platform / configuration
  - For very large databases, the Backup Context can consume a lot of memory and overflows shared memory
  - Sample Crash Callstack
    - @[ 8] 0x6017aca8 nnotes.Panic@4+520 (60bb0c4f)
    - @[ 9] 0x6017ad2c nnotes.Halt@4+28 (107)
    - **@[10] 0x60103e95 nnotes.AccessAllProtected@0+85 ()**
    - **@[11] 0x600469fe nnotes.AccessAll@8+46 (1,1)**
    - @[12] 0x60047a83 nnotes.ProcessGlobalEvent@4+19 (1512ee4)
- Limit the amount of backup memory used
  - Block Type: 0x02e9 check TN #1211241 for details
  - Notes.ini
    - NSF_Backup_Memory_Constrained=1 (defaults to 20 MB)
    - NSF_Backup_Memory_Limit=200000000 (reasonable size: 200 MB)

# Lotus Domino Statistics

- Valuable resource of information
  - Combines Domino Statistics and Platform statistics
  - Platform statistics depend on the OS platform but are sort of unified between platforms
    - Check events4.nsf for a description of each platform stat on each platform
  - You should collect Server stats at least every 15 minutes (default is 90 minutes)
  - Configure statistic events for important stats with the right thresholds
    - Keep long term data to compare current and historic data
  - You can also leverage SNMP to query stats
    - Limitation: Only works for one partition per OS instance on all platforms
  - C-API allows you to add own stats and also some Domino Probes generate own stats
- For a whole session about logging and statistics check
  - Lotusphere 2008 / BP112 In the Land of the Blind, Logs Make You King
    - http://www.nashcom.de/lotusphere

# Client Clocking

- Can be used to track Notes Client/Server Transactions (NRPC)
  - Logs
    - transaction name
    - transaction data
    - response time (ms)
    - bytes send, received
- Example:
  - (15-78 [15]) OPEN_NOTE(REPC1256B16:0072BCBE-NT00000E3E,00400020): 0 ms. [52 +1454=1506]
- Enable on client via
  - client_clock=1
  - debug_console=1
    - Enables a debug text window -- never close this manual, causes a crash
  - debug_outfile=c:\debug_notes.log
    - Writes a debug log file

# Selected Transaction Types

| | |
|---|---|
| START_SERVER | Start User Session |
| OPEN_DB/CLOSE_DB | Open/Close a database |
| OPEN_NOTE/NIF_OPEN_NOTE | Open a Note |
| UPDATE_NOTE | Update a Note – there is no close transaction |
| OPEN_COLLECTION/ CLOSE_COLLECTION | Open/Close a view/folder collection |
| READ_ENTRIES | Reads data from a view/folder |
| UPDATE_COLLECTION | Updates a view/folder collection |
| FIND_BY_KEY | Finds notes in a view/folder collection |
| FINDDESIGN_NOTES | Finds design notes |
| SEARCH | Search operation with formula |
| GET_MODIFIED_NOTES | Find table of modified notes |
| GET_ALLFOLDERCHANGES_RQST | Get changes in all folders<br>Unread count in mail folders |

# Selected Transaction Types

| | |
|---|---|
| NAME_LOOKUP | Lookup information in Domino Directory |
| ALLOC_UPDATE_OBJECT | Create or update an object |
| READ_OBJECT | Read data from an object |
| WRITE_OBJECT | Write data to an object |
| READ REPLICATION HISTORY | Read Replication History |
| DB_INFO_GET | Get database info buffer |
| GET_NOTE_INFO | Get Note information |
| DB_MODIFIED_TIME | Get the modified date of DB |
| DB_REPLINFO_GET | Get Replication info |
| POLL_DEL_SEQNUM | Get Delivery Sequence Number |
| DB_GETSET_DEL_SEQNUM | Get or set Delivery Sequence Number |
| GET_SPECIAL_NOTE_ID | Get a special Note-ID by number |

# Nagle Algorithm on Unix/Linux

- The Nagle Algorithm (John Nagle) is designed to optimize small packets (like for telnet sessions)
  - Small packets are combined to larger packets – when the next packet to client is sent
  - In case of Domino this can cause delays because individual transactions might be delayed
    - The wait time is up to 200 ms!
    - Notes transactions are serialized, no other packets need to be send to client at the same time
  - The Nagle Algorithm should be disabled for Domino on all Unix/Linux platforms
    - Disabled by default in 8.5, Previous releases, notes.ini setting: **debug_pd_nagle_off=1**

- Example: 1000 Note Open Transactions
  - With Nagle Algorithm
    - Average: **62 ms**
  - Disabled Nagle Algorithm
    - Average: **5 ms**
  - Results in detail / Response time distribution
    - Before / After -->

```
[   0] ->  547
[  15] ->   55
[  16] ->  102
[  31] ->   23
...
[ 203] ->   15
[ 218] ->   71
[ 219] ->   48
[ 234] ->   59
[ 249] ->   12
[ 250] ->   15
[ 265] ->   10
...
```

```
[   0] ->  658
[  15] ->  125
[  16] ->  192
[  31] ->   15
[  32] ->    5
[  47] ->    3
[  62] ->    1
[  78] ->    1
```

# Current Issue: Broken Design Collection

- Domino has an internal design cache in each database to find design notes
  - Used by NIFFindDesignNoteExt (Transaction: FINDDESIGN_NOTES)
- In some odd cases the design cache breaks
  - Without the design cache the client tries to find design elements the “old style” by opening and searching the design collection.
  - This causes quite a bit overhead – specially for WAN connections
- Design Collections is discarded when the internal cache table overflows
  - Happens when 40 or more design elements have the same name
    - This happens regularly with private on first use folders/view
    - Only work-around: Avoid private on first use folders/views and remove existing folders
  - Reference: SPR #RSTN7K2EM4, TN #1322578 Performance degradation using "Private on First Use" views or folders

# Borken Design Collection Client_Clock Data Example

```
(3-299 [309]) OPEN_COLLECTION(REPC1256E62:004B8651-NTFFFF0020,0000,0000): 0 ms. [42+34=76]
(4-299 [310]) READ_ENTRIES(REPC1256E62:004B8651-NTFFFF0020): 0ms. [76+65148=65224]
(5-299 [311]) READ_ENTRIES(REPC1256E62:004B8651-NTFFFF0020): 16 ms. [76+65388=65464]
(6-299 [312]) READ_ENTRIES(REPC1256E62:004B8651-NTFFFF0020): 0ms. [76+63918=63994]
(7-299 [313]) READ_ENTRIES(REPC1256E62:004B8651-NTFFFF0020): 0ms. [76+65506=65582]
(8-299 [314]) READ_ENTRIES(REPC1256E62:004B8651-NTFFFF0020): 0ms. [76+65498=65574]
(9-299 [315]) READ_ENTRIES(REPC1256E62:004B8651-NTFFFF0020): 15 ms. [76+64920=64996]
(10-299 [316]) READ_ENTRIES(REPC1256E62:004B8651-NTFFFF0020): 0 ms. [76+63688=63764]
(11-299 [317]) READ_ENTRIES(REPC1256E62:004B8651-NTFFFF0020): 0 ms. [76+64522=64598]
(12-299 [318]) READ_ENTRIES(REPC1256E62:004B8651-NTFFFF0020): 0 ms. [76+65402=65478]
(13-299 [319]) READ_ENTRIES(REPC1256E62:004B8651-NTFFFF0020): 0 ms. [76+65478=65554]
(14-299 [320]) READ_ENTRIES(REPC1256E62:004B8651-NTFFFF0020): 16 ms. [76+65296=65372]
(15-299 [321]) READ_ENTRIES(REPC1256E62:004B8651-NTFFFF0020): 0 ms. [76+65238=65314]
(16-299 [322]) READ_ENTRIES(REPC1256E62:004B8651-NTFFFF0020): 0 ms. [76+65386=65462]
(17-299 [323]) READ_ENTRIES(REPC1256E62:004B8651-NTFFFF0020): 0 ms. [76+9480=9556]
(18-299 [324]) CLOSE_COLLECTION(REPC1256E62:004B8651-NTFFFF0020): 0 ms. [12+0=12]
```

```
((2-507 [330]) FINDDESIGN_NOTES: 31 ms. [42+16=58]
```

# Server_Clock & Show Trans

- The server keeps track of all transactions
  - Also used for LOADMON (part of SAI calculation)
- You can display transaction summary via "**show trans**"
  - And reset the summary counters via "**show trans reset**"
- Or you can display transactions via console log via **server_clock=1**
- **Server_clock=1** has some limitations
  - Only shows transaction information but no user or database information
  - **38965515 ms 'OPEN_DB' 0 ms (0 ms NETIO) TCPIP 000403B1 Rcvd 0 Sent 216**
- New server_clock options have been introduced in Domino 8.5.1
  - Have first been implemented thru hotfixes for SAI troubleshooting and finally helped fixing SAI :-)

# Additional Server_Clock in Domino 8.5.1

- Server_Clock=2
  - Will dump more information
  - Username, Database, IP Address, and if transaction is used for LOADMON (Lm 1)

```
 39255671 ms 'OPEN_DB' 0 ms (0 ms NETIO) TCPIP 000403B1 Rcvd 0 Sent 254
User 'Daniel Nashed/NashCom/DE' Db 'acl.nsf' Ip '192.168.100.3' Lm 1
```

- Server_Clock=3, DEBUG_TRANSACTION_TIME=n
  - Dumps only transaction taking longer than the specified time
  - Can help to reduce the number of transactions dumped and only lists “slower” transactions
  - For example:  5000 ms
    - Take care: But some transaction like open view collections might take longer than 5 seconds without indicating a problem

# SAI and LOADMON

- **Domino uses a module called "LoadMon"**
  - Routine calculating speed of 12 selected transactions
  - Checks current transaction performance, summarizes and compares them with previous intervals and minimum values (RunningAvgTime & MinAvgTransTime)
  - Unit: microseconds

- OPEN_DB
- OPEN_NOTE
- CLOSE_DB
- DB_INFO_GET
- DB_REPLINFO_GET
- GET_OBJECT_SIZE
- READ_OBJECT
- GET_SPECIAL_NOTE_ID
- DB_READ_HIST
- DB_WRITE_HIST
- SERVER_AVAILABLE_LITE
- NIF_OPEN_NOTE

# Expansion Factor (XF)

- XF is calculated based on the performance values of current transactions in relation to minimum time for a transaction
  - It's the number of times the current transactions take longer than the minimum transaction time
  - XF values for different transactions build a overall XF
  - This XF is computed and converted into AI based on a Range to scale the XF
- SAI is calculated based on XF and the transinfo range (n)
- SAI = 100 * (1 - log (XF) / log (2) / n)
  - Notes.ini Server_Transinfo_Range n is 6 by default and specifies the maximum Expansion Factor of a Domino Server. The XF is calculated 2 raised to the power n (64 by default)

# Issues with SAI and LOADMON

- SAI was broken until Lotus Domino 8.5
  - There have been a couple of issues with LOADMON. The last one known has been fixed in D8.5
- SAI calculation on fast servers still might not work for you out of the box
  - LOADMON uses micro seconds
    - On a fast server at idle times transactions can take only a couple of micro seconds
    - Compared to normal performance e.g. 1 ms this can result in very high **XF**
    - Causes a low **SAI** for normal performing servers
- Tuning: D8.5 Set range of minimum and maximum values
  - notes.ini: **Server_MinPossibleTransTime**=1500
  - notes.ini: **Server_MaxPossibleTransTime**=20000000
  - Important: You have to delete loadmon.ncf when the server is shutodwn to delete old minimum values
- If LOADMON is configured correctly SAI can help to measure performance
  - E.g. Set the **Server_MinPossibleTransTime** to your expected response time

# More LoadMon Notes.ini Settings

- SERVER_TRANSINFO_MAX (default 5 / max 60)
  - number of statistics collections stored in LoadMon
- SERVER_TRANSINFO_UPDATE_INTERVAL (default 15)
  - interval for statistics capturing & calculation
- SERVER_MIN_TRANS (default 5)
  - minimum transactions needed for a statistic value to be valid
- SERVER_TRANSINFO_NORMALIZE (default 3000)
- SERVER_TRANSINFO_HTTP_NORMALIZE (12000)
  - used to initialize empty statistics (zero in loadmon.ncf) on startup in Domino

# Debugging LoadMon

- debug_loadmon=1
  - Enables LoadMon Debugging, writes additional information to server console
  - Loadmon: Domino AI = 100, XF = 1
  - Adds additional 46 statistics counters (server.loadmon.*)
- loadmon.ncf
  - loadmon.ncf in Domino data directory stores last information from loadmon before server is shutdown
  - loaded on server start to initialize statistics counters

# Lotus Domino 8.5 I/O Statistics

- Domino records I/O data per process and database
  - Process Name, Database Name
  - NumFileWrites, NumFileReads
  - MBWritten, MBRead = MB written/read
  - NoteOpens, ProfileNoteOpens, DesignNoteOpens
  - NTUpdateAdd, NTUpdateUpdate, NTUpdateDelete
  - NTUpdateExpiredSoftDeletes, ProfileNoteUpdates, DesignNoteUpdates
- Provides detailed information about I/O operations in CSV Format
  - Show iostat writes a file into the IBM_TECHNICAL_SUPPORT directory

```
> sh iostat
[0AA4:0007-05DC] IOSTAT dumped to file C:\Lotus\Domino85\data\IBM_TECHNICAL_SUPP
ORT\iostat_nsh-d85-win-01_2009_12_20@11_15_18.csv
^
```

```
"nupdate","",327,6056,1,17,648,19,481,0,11,0,0,0,11
"","C:\Lotus\Domino85\data\statrep.nsf",1351,32266,10,160,8314,0,4406,183,35,0,0,0,142
```

# Summary

- There are a lot of diagnostic features in Lotus Notes/Domino
  - Some features are designed for crash and failure analysis
    - There is much more than just NSD and Fault Recovery
  - Domino 8.x also has many features to troubleshoot performance issues on client and server side
- A 60 minute session can only give you ideas what to look into
  - Many areas could be a complete separate session
  - This session should give you ideas what to look for
  - And to help understand why IBM support is asking for certain data
- Not all troubleshooting information is easy to understand
    - Some is build from developers for developers ...

# Links and Resources

- Technotes
  - TN #7007508 - Knowledge Collection: NSD for Notes Domino release 6 and 7
    - Many of those documents are still relevant for Domino 8.x/8.5.x
  - TN #4013182 - Updated NSD for Domino releases

- Lotus Developer Domain
  - http://www.ibm.com/developerworks/lotus

- Also check Knowledge Base and Fixlist Database

# Q&A

- I hope you enjoyed the session
- Please fill out your evaluations!
- Questions?
- Presentation Updates
  - http://www.nashcom.de/lotusphere
- Contact
  - nsh@nashcom.de
  - http://www.nashcom.de
  - http://blog.nashcom.de

# Legal Disclaimer



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 presentation 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 presentation 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. Nothing contained in these materials is intended to, nor shall have the effect of, stating or implying that any activities undertaken by you will result in any specific sales, revenue growth or other results.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput or performance that any user will experience will vary depending upon many factors, including considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve results similar to those stated here.