"CSI Domino"

Diagnostic Collection & NSD Analysis

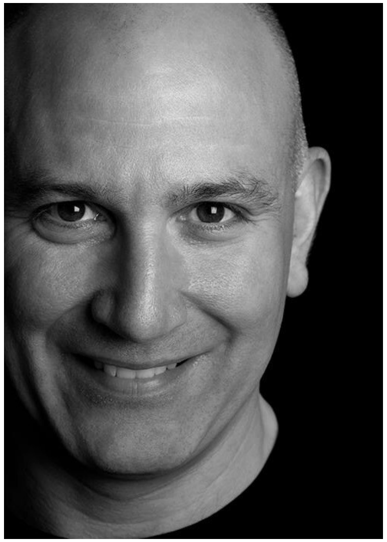
Daniel Nashed

AdminCamp 2019 – 25-27 March in Gelsenkirchen

Nash!Com
Communication Systems

About the presenter

- **Nash!Com – German IBM® 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, IBM® Domino® Infrastructure, Administration, Integration, Troubleshooting and IBM® Traveler**
  - Platform Focus: Microsoft® Windows® 64, Linux® and IBM AIX®
- **Author of the Domino on Linux®/UNIX® Start Script**
  - Note: Working on RHEL7 + SLES 12 "systemd" support



IBM CHAMPION

Nash!Com
Communication Systems

Agenda

- **Introduction – What is „Serviceability“**
- **Automatic Data Collection (ADC), Configuration Collector**
- **NSD, Memcheck**
  - Server Crashes, Hangs, Annotation of NSDs
- **Memory Management**
- **Advanced Methods**
  - Semaphore Debugging
  - Memory Dumps
- **Performance Troubleshooting**
- **Q&A – Any time**

Useful Software & Tools

- **Software**
  - Notes Peek
  - Lotus Notes Diagnostics (LND)
  - 7Zip – open source ZIP tool
  - Ultraedit (commerical but great) or Notepad++ (free)

  - NashCom Tools
    - nshcrash
    - Nshmem
- **C-API Toolkit**
  - Great source of information

What is Serviceability?

- **RAS = Reliability Availability Serviceability**

- **RAS is the effort to improve the 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 to older releases

- **It's not just about NSD & Memcheck but all parts of Domino**
  - Logging, Debug Options, etc

- **Great help for Admins, Developers and Troubleshooters**

Diagnostic Features in Domino

- **Directory \IBM_TECHNICAL_SUPPORT**
  - Single place of log files collection
  - Many many debug options
- **Dynamical Console Log**
  - Log file containing all log information
- **Automatic Data Collection / Configuration Collector**
  - Server and Client mail self-acting, configuration snap-shot
- **Debug Options on Server and all Servertasks**
- **Domino Domain Monitoring (DDM)**
  - Comprehensive Server Monitoring

Diagnostic Features in Domino

- **NSD**
  - Notes System Diagnostics
- **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**
    - D8: Improved crash info contained within e-mail notification

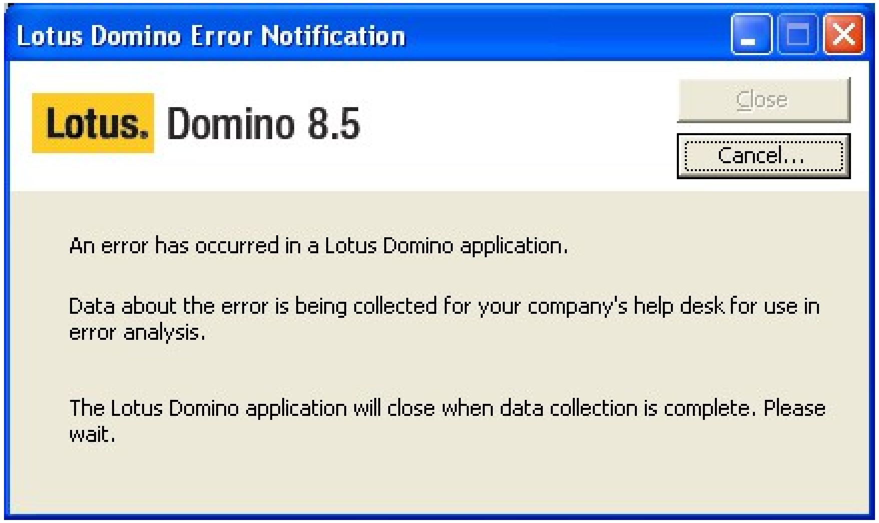
- **How Fault Recovery works**
  - Run NSD if configured
  - Cleans up resources
  - Restarts Server

Transaction Logging

- **Recommended for all Domino server types**
  - Changes the way databases are locked for concurrent access
    - Lock-Manager optimizes performance
    - Changes are written sequentially into translog
    - Asynchronous Log Manager writes data into databases afterwards
      - Process can continue to run meanwhile
  - Without Transaction Logging, databases open at crash time are inconsistant
    - Needs fixup for all open databases which causes load on server and delays the time until Domino Server is completely back only after a crash
    - Data Loss possible without Transaction Log in crash situations
  - With transaction logging the recovery operation writes pending changes into the open databases at crash time
- **Recommend: Domino Backup API aware Backup Solution**
  - Or shutdown your Domino Server for backup at night
  - All other backup operations are completely unsupported

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 in one central repository.**
  - Senddiag servertask runs on startup to collect information like NSDs
- **Server Configuration Doc / Diagnostics Tab**
  - Fault-Report Database (lndfr.nsf) as Mail-in Database
    - Size for diagnostic data, retention days, ...
    - Filter pattern to add to data collection (file-patterns!)
- **ADC and Fault Recovery is also available for Notes Clients**
  - Configured in desktop policy
- **Mail Size Limit in Config Document also applies for diag mails**
  - For Clients and Servers



An error has occurred in a Lotus Domino application.

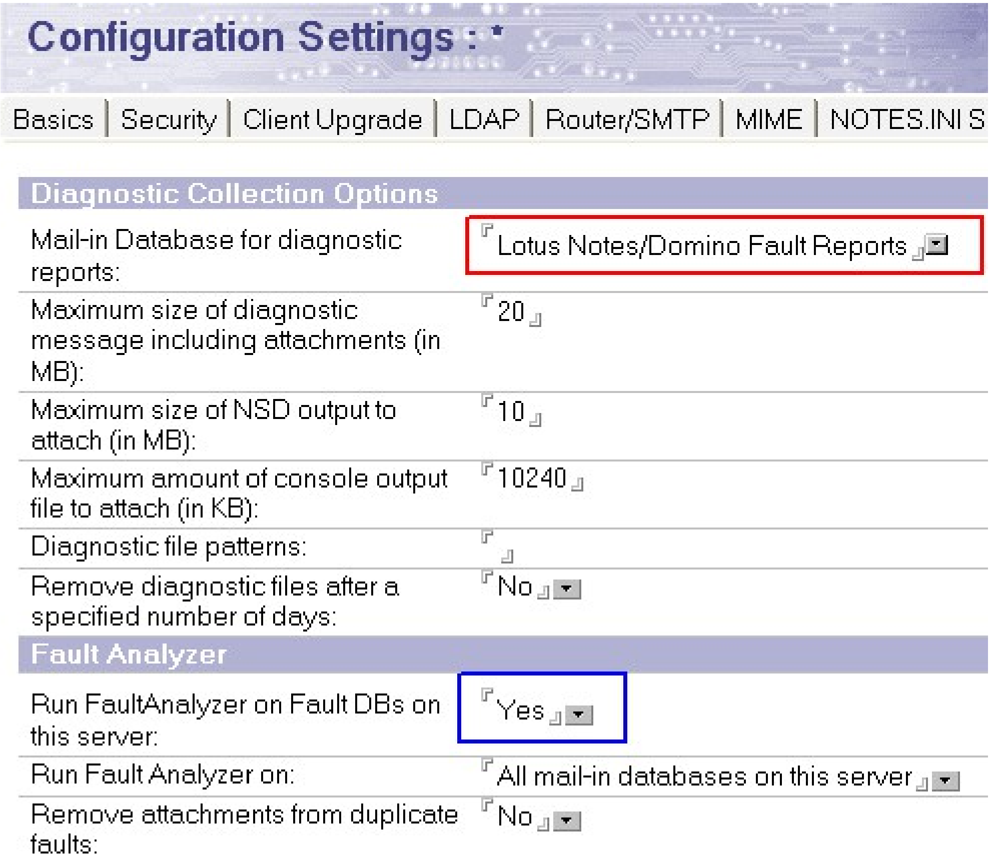
Data about the error is being collected for your company's help desk for use in error analysis.

The Lotus Domino application will close when data collection is complete. Please wait.

Fault Analyzer Servertask

- **Fault Report Database is typically located on admin server**
  - Mail-In Database with Fault Report Template

- **Fault Analyzer Servertask**
  - Used to annotate, categorize NSDs (since R7)
    - Similar call-stacks, Same Domino releases, Client or Server
  - You should have separate databases for clients / servers



Diagnostic Collection Options

Mail-in Database for diagnostic reports:	Lotus Notes/Domino Fault Reports
Maximum size of diagnostic message including attachments (in MB):	20
Maximum size of NSD output to attach (in MB):	10
Maximum amount of console output file to attach (in KB):	10240
Diagnostic file patterns:	
Remove diagnostic files after a specified number of days:	No

Fault Analyzer

Run FaultAnalyzer on Fault DBs on this server:	Yes
Run Fault Analyzer on:	All mail-in databases on this server
Remove attachments from duplicate faults:	No

Configuration Collector

- **Provides snapshots of how a Domino server is configured**
  - Located in IBM_TECHNICAL_SUPPORT directory
  - Configuration files
    - Server Document (serverdoc_<server>_<date>_<time>.dxl)
    - Configuration Document (configall_<server>_<date>_<time>.dxl)

    - Format: DXL – Domino XML Format
      - Tip: Can be imported back into a Domino Directory
        You can use the dxlimport example form the C-API toolkit ;-)
  - Sysinfo NSD (sysinfo_<server>_<date>@<time>.log)
    - Contains information about environment
      - Notes.ini, System Environment (details later)

Dynamic Console Log

- **Contains all logging information**
  - Including debug information
  - Same as notes.ini debug_outfile!
- **Server commands**
  - start consolelog / stop consolelog / sh server
- **Tip: By default the console log file shunk size is 10 MB**
  - Change via notes.ini Console_Log_Max_Kbytes=n
    - Used to be 1 KB only in earlier releases
    - Or completely enable console log
      - This will keep all console log data
- **Only needed for Windows**
  - On Linux/Unix you can use the console out redirection

Domino Server Controller

- **Needed since Win2008 because of Security change**
  - Even server is using the system account a native console window is not allowed
  - Workaround: Using the Server Controller
    - Automatically installed on Win2008 and higher
  - Also useful in other types of environment – specially windows
  - Reduces the need for remote control software and direct OS level access
- **Start Server Controller by changing the nserver parameter**
  - nserver -jc instead of nserver
- **Java Controller can be used cross platform to connect to the server console**
  - Username/Password is needed
  - OS-Level commands can be executed
  - Server can be killed and started remotely

NSD - Notes System Diagnostics

- **Has been around for years in Domino**
  - Fully available since a very long time (Domino 6.0 for Win32)
  - Replaced the old RIP in Domino 6 for Win32
    - Not a „Just in Time“ (JIT) Debugger

- **It's invoked automatically if Server/Client crashes**
  - Or you can manually invoke it for troubleshooting

- **NSD provides a huge collection of system diagnostics information on Domino and Operating System level**

- **Used by Admins, Developers and Support for Troubleshooting**

NSD - Startup

- **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!!!

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

Run NSD as a Service

- **New Feature since Domino 8 allows NSD to run as a service**
  - Avoids issues with 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

- **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

NSD Help Files in Domino 8

- **Check data/help directory for NSD documentation**
- **nsddoc.html**
  - Main entry point for documentation
- **nsdcmds.html**
  - NSD commands
- **nsdini.html**
  - nsd.ini options
- **nsdopts.html**
  - NSD options
- **memcheck.html**
  - Memcheck documentation (not yet available in D8.0 Gold)

Why Server Freeze and Server Panic?

- **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

- **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**
- **Non-Core/Third Party code**
  - DECS/LEI, Oracle, DB2, JDBC, etc.
- **Corrupt data**
  - Corrupt documents, etc …
- **Memory Management issues**
  - Overwrites, handle locking, memory leaks)
- **Insuffcent 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
- **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

C-API Toolkit

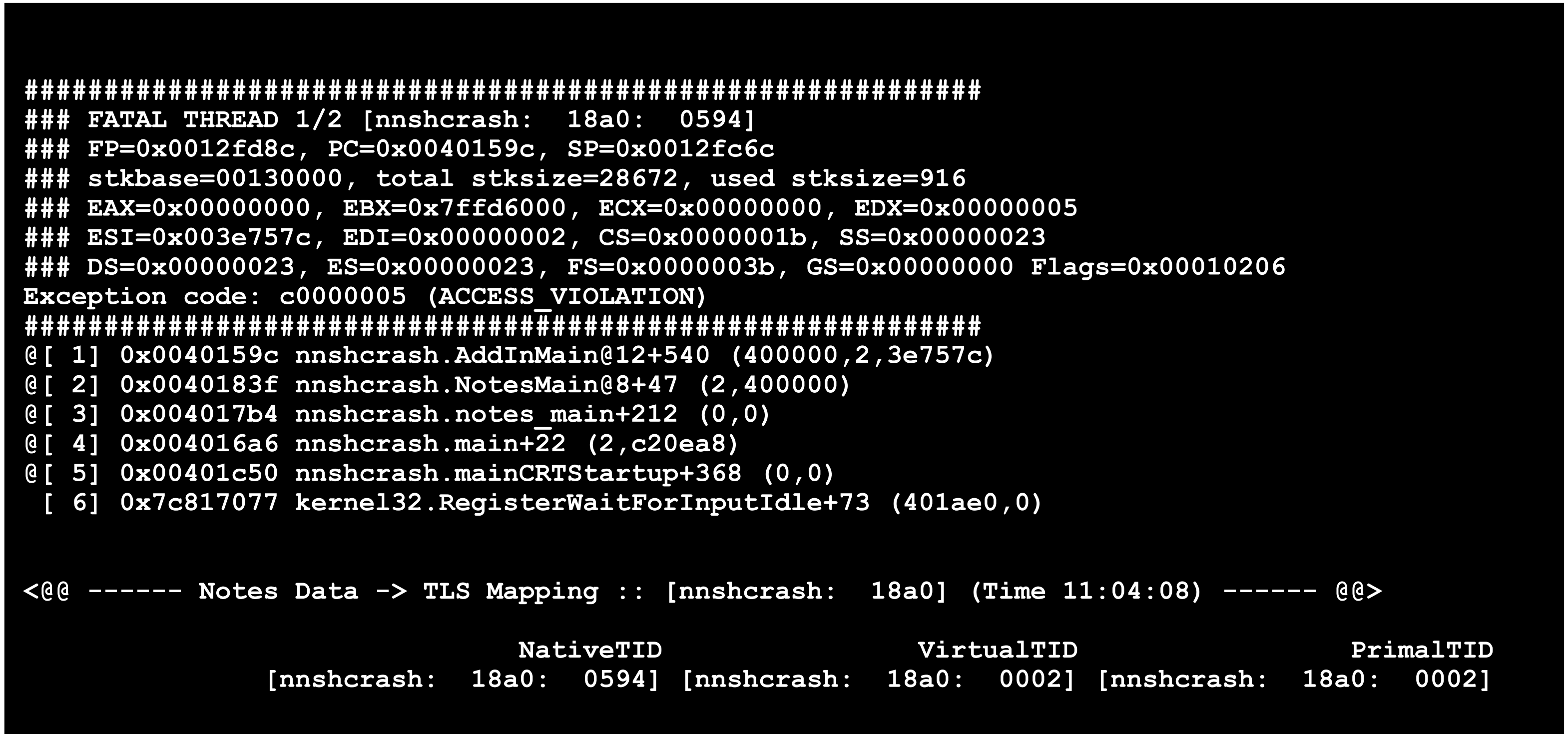
- **The C-API Toolkit contains a sub-set of the internal APIs used by Notes Development to build Notes/Domino**
  - Can be used to build your own servertasks, extension-manager, DSAPI filters, client applications
- **Also a great resource for finding information about Domino**
- **Components**
  - Header files
    - extract from the original Notes/Domino code
  - Reference Database
    - Documentation for all exposed calls
  - User Guide Database
    - Documentation how to use the toolkit and information about Notes/Domino internal Architecture etc.
  - Sample Applications

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

How to find affected databases?

- **Check the Physical Virtual Thread Mapping**
  - To find VTHREAD of crashing process/thread for open databases



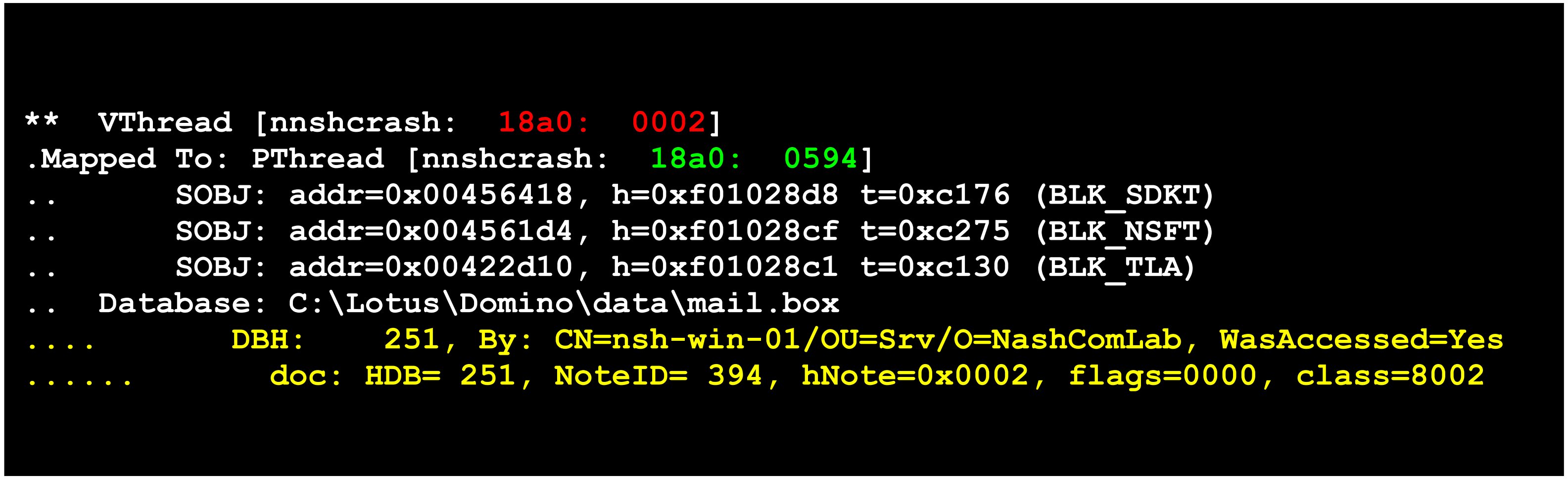
```
############################################################
### FATAL THREAD 1/2 [nnshcrash:  18a0:  0594]
### FP=0x0012fd8c, PC=0x0040159c, SP=0x0012fc6c
### stkbase=00130000, total stksize=28672, used stksize=916
### EAX=0x00000000, EBX=0x7ffd6000, ECX=0x00000000, EDX=0x00000005
### ESI=0x003e757c, EDI=0x00000002, CS=0x0000001b, SS=0x00000023
### DS=0x00000023, ES=0x00000023, FS=0x0000003b, GS=0x00000000 Flags=0x00010206
Exception code: c0000005 (ACCESS_VIOLATION)
############################################################
@[ 1] 0x0040159c nnshcrash.AddInMain@12+540 (400000,2,3e757c)
@[ 2] 0x0040183f nnshcrash.NotesMain@8+47 (2,400000)
@[ 3] 0x004017b4 nnshcrash.notes_main+212 (0,0)
@[ 4] 0x004016a6 nnshcrash.main+22 (2,c20ea8)
@[ 5] 0x00401c50 nnshcrash.mainCRTStartup+368 (0,0)
 [ 6] 0x7c817077 kernel32.RegisterWaitForInputIdle+73 (401ae0,0)


<@@ ------ Notes Data -> TLS Mapping :: [nnshcrash:  18a0] (Time 11:04:08) ------ @@>

                               NativeTID                VirtualTID                 PrimalTID
               [nnshcrash:  18a0:  0594] [nnshcrash:  18a0:  0002] [nnshcrash:  18a0:  0002]
```

How to find affected databases?

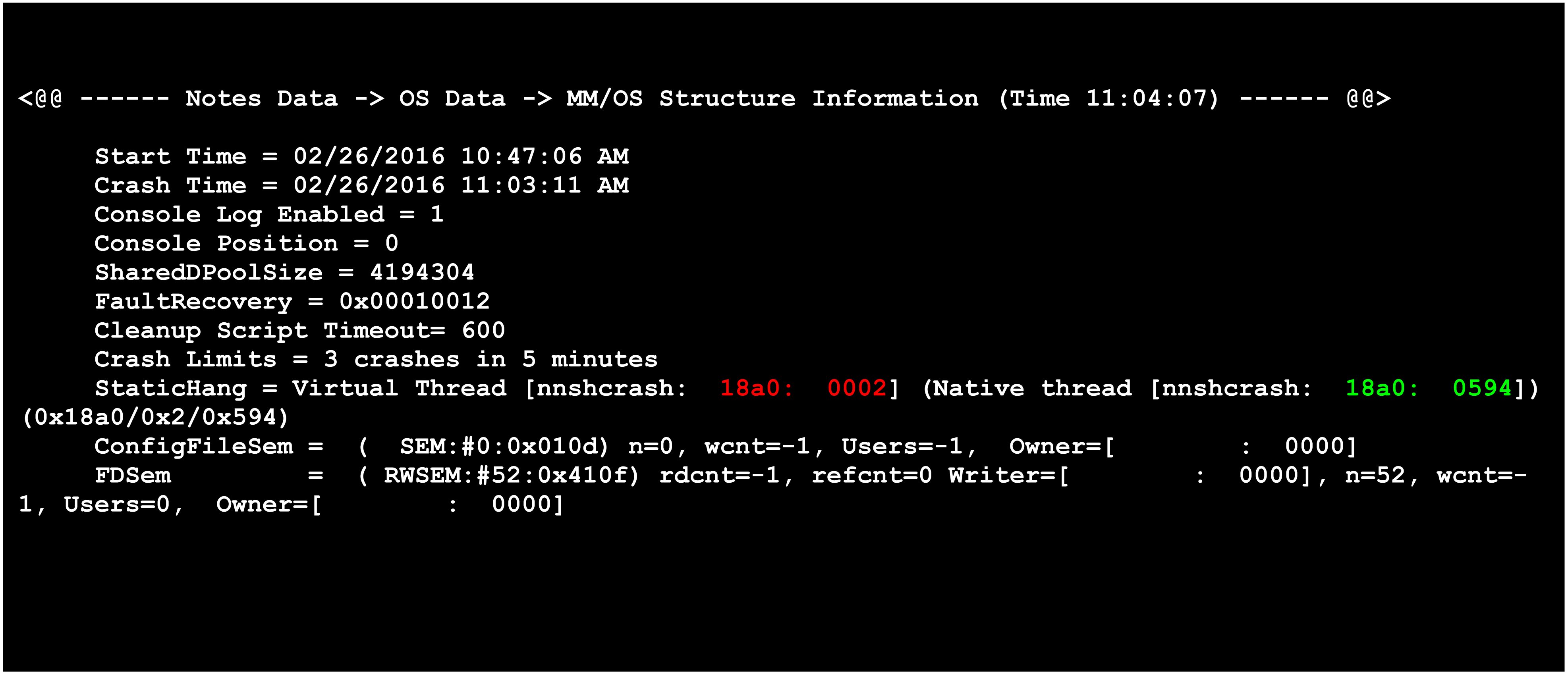
- **VTHREAD contains all open databases and notes**
  - Notes can be documents, profile docs or design



```
**  VThread [nnshcrash:  18a0:  0002]
.Mapped To: PThread [nnshcrash:  18a0:  0594]
..      SOBJ: addr=0x00456418, h=0xf01028d8 t=0xc176 (BLK_SDKT)
..      SOBJ: addr=0x004561d4, h=0xf01028cf t=0xc275 (BLK_NSFT)
..      SOBJ: addr=0x00422d10, h=0xf01028c1 t=0xc130 (BLK_TLA)
..  Database: C:\Lotus\Domino\data\mail.box
....       DBH:    251, By: CN=nsh-win-01/OU=Srv/O=NashComLab, WasAccessed=Yes
......       doc: HDB= 251, NoteID= 394, hNote=0x0002, flags=0000, class=8002
```

How to find affected databases?

- **It's not simple on all platforms to match physical /logical (virtual) threads**
  - OS Data -> MM/OS Structure Information can help to find VTHREAD
  - Tip: Look for the exact string “**StaticHang =**”



```
<@@ ------ Notes Data -> OS Data -> MM/OS Structure Information (Time 11:04:07) ------ @@>

     Start Time = 02/26/2016 10:47:06 AM
     Crash Time = 02/26/2016 11:03:11 AM
     Console Log Enabled = 1
     Console Position = 0
     SharedDPoolSize = 4194304
     FaultRecovery = 0x00010012
     Cleanup Script Timeout= 600
     Crash Limits = 3 crashes in 5 minutes
     StaticHang = Virtual Thread [nnshcrash:  18a0:  0002] (Native thread [nnshcrash:  18a0:  0594])
(0x18a0/0x2/0x594)
     ConfigFileSem =  (  SEM:#0:0x010d) n=0, wcnt=-1, Users=-1,  Owner=[        :  0000]
     FDSem         =  ( RWSEM:#52:0x410f) rdcnt=-1, refcnt=0 Writer=[        :  0000], n=52, wcnt=-
1, Users=0,  Owner=[        :  0000]
```

How to match affected databases?

- **To identify the DB causing the crash**
  - Search the Call-Stack for Database Handles and NoteIDs
    - e.g. NSFNoteOpen(DBHANDLE hDb, NOTEID NoteID, WORD flags, NOTEHANDLE *hNote);
- **A handle (DBHANDLE) is represented by a hex number in the call stack**
  - Can be found in open database list
  - Take care: Handle number in open database list is decimal !
  - A NOTEID is also a hex value which identifies a Note in a Database
- **Seach for "NoteID"**
  - Either Admin Client Database Tools
  - Notes Peek
  - Your favorite other ISV AdminTool

NotesPeek – Still a very useful Tool!

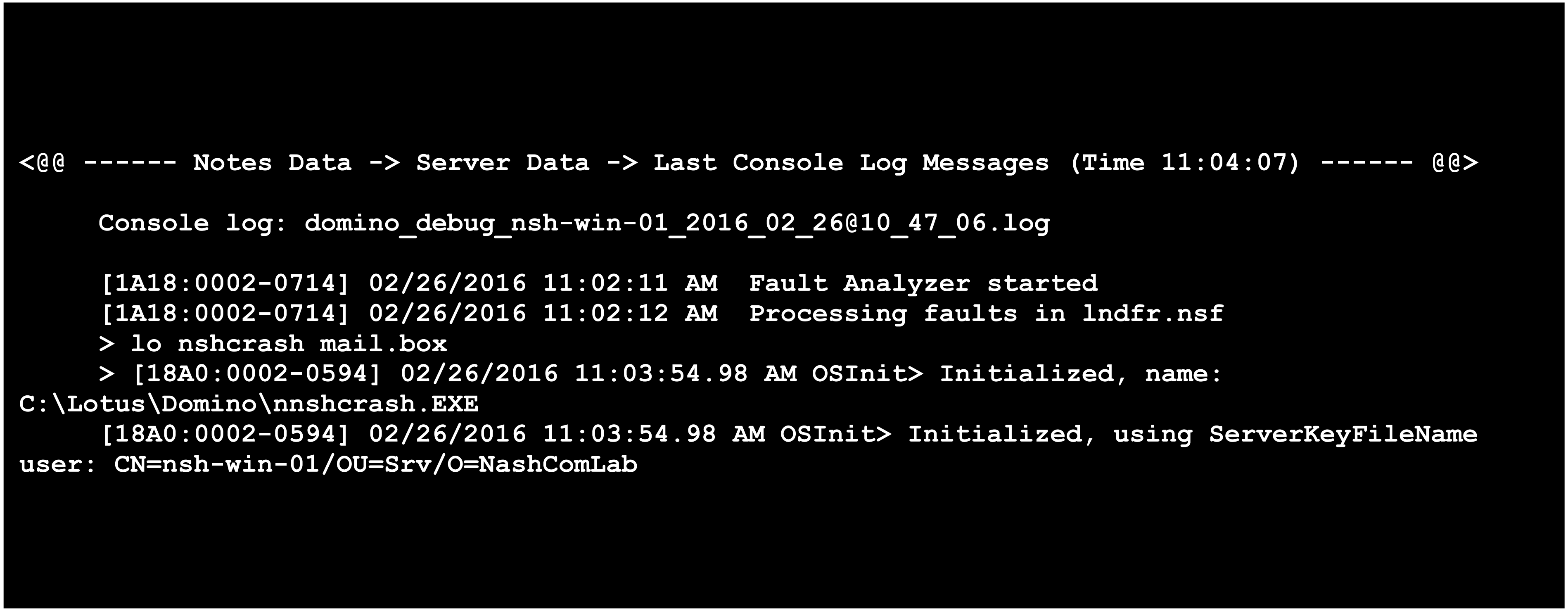
- **NotesPeek is a free application developed by Lotus Development which can be downloaded for free**

- **Allows you to peek into Notes Databases in a very low way**
  - Finding notes by NoteID, UNID etc.
  - Opening Profile Documents
  - Find deletion stubs
  - Look into each field including richtext and CD records

- **Download Link NotesPeek 1.53**
  - http://www.ibm.com/support/docview.wss?uid=swg24005686
  - Quite old version but still works with Notes 9.0.1

Last Lines from Console Log

- **Current versions of MEMCHECK contain last console log lines**
  - That's why console logging has been changed
  - Very useful to figure out what last happened on the server



```
<@@ ------ Notes Data -> Server Data -> Last Console Log Messages (Time 11:04:07) ------ @@>

     Console log: domino_debug_nsh-win-01_2016_02_26@10_47_06.log

     [1A18:0002-0714] 02/26/2016 11:02:11 AM  Fault Analyzer started
     [1A18:0002-0714] 02/26/2016 11:02:12 AM  Processing faults in lndfr.nsf
     > lo nshcrash mail.box
     > [18A0:0002-0594] 02/26/2016 11:03:54.98 AM OSInit> Initialized, name:
C:\Lotus\Domino\nnshcrash.EXE
     [18A0:0002-0594] 02/26/2016 11:03:54.98 AM OSInit> Initialized, using ServerKeyFileName
user: CN=nsh-win-01/OU=Srv/O=NashComLab
```

More Information - Open Files/Documents

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

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
  - Process monitor or on Unix/Linux: ChildDied Signal terminates server
- **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
  - **DEBUG_THREADID=1**
    - Logs thead-id for every log output

Next Steps

- **Customer can only fix data problems, check/add server resources (e.g. memory) or install later versions**

- **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 source code**
  - Take care: NSD also contains some sensitive information about your system and users.
    - Check the NSD before sending it to external people

SYM File Support for Add-On Products

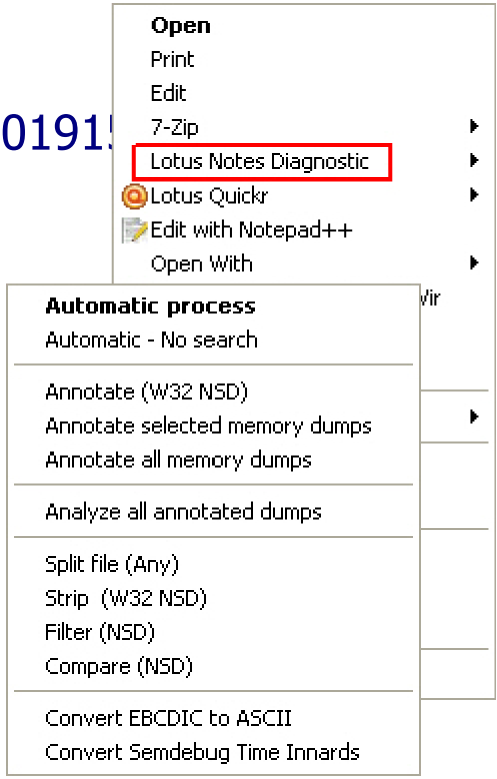
- **Domino uses a special SYM file format integrated into one large SYM file**
  - Since D6.5.1 Domino is able to read SYM files for individual binaries
  - For previous versions keep debugging code in your applications to get proper annotated call-stack for 3rd party products
  - Microsoft mapsym cannot be used to generate sym files for Notes/Domino
- **Lotus Development (Iris) Tool Map2iSym is part of the Lotus C-API Toolkit since Domino 6.5.1**
  - Ability for NSD to integrate 3rd party "Domino family products"
  - Starting with D6.5.1 NSD it works also extended Domino products
- **Since Domino 9 64bit “PDB” files are used instead of SYM**
  - Only IBM uses special formated SYM files
  - No new Map2iSym available but you can use stripped PDB files

Lotus Notes Diagnostics (LND)

- **Tool to annotate NSDs, semdebug files, memory dumps etc**
  - Current Version 2.9 downloadable from IBM
  - http://www.ibm.com/support/docview.wss?rs=899&uid=swg240191

- **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



Domino Memory Management

- **Domino uses an own Memory Management Layer**
  - Different Memory types
    - Pooled memory (DPOOLS)
    - Direct memory allocations
  - Local and Shared Memory
    - Shared Memory for all Servertasks
    - Local Process Memory per tasks
  - 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

Memory Limitations

- **Domino has only a certain amount of addressable memory for**
  - Local Memory – separate for each process
  - Shared Memory – shared between all Domino processes

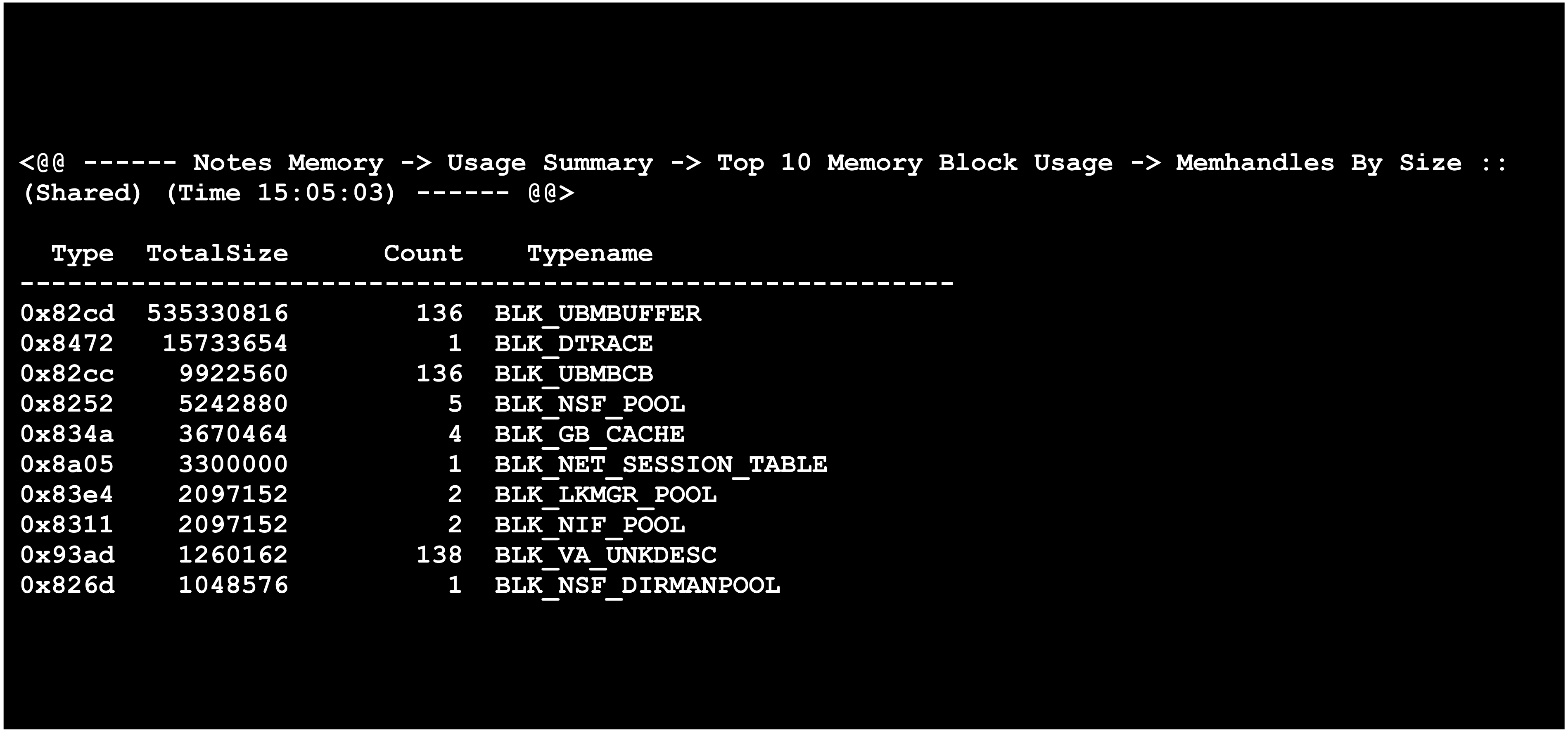
- **The limit depends on the platform**
  - Combination of shared memory + local process memory is the limiting factor
  - For 32bit the total limit is 4GB at most
  - The larger part of memory used is shared memory

- **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

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 Dumps

- **You can dump memory**
  - Run "**server -m**"
  - Or „**show memory dump**"
  - Program document every hour: **nserver.exe**
    Parameters: **-c "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

Memory Trap Leak Debugging

- **Once you figured out about a problematic Memory Block Type you can enable Trap Leak Debugging**
  - Debug_Trapleaks=0x**3A45**
    - For shared memory ensure that you take care of the shared memory bit **0x8000** – the bit must be removed from the value
  - Debug_Trapleaks_ShowStack=1
  - DEBUG_SHOWLEAKS=1
  - DEBUG_DUMP_FULL_HANDLE_TABLE=1
  - DEBUG_DUMP_BLOCKCODES=1
- **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/config
  - 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 on Domino 32Bit**
  - Block Type: 0x02e9 check TN #1211241 for details
    - NSF_Backup_Memory_Constrained=1 (defaults to 20 MB)
    - NSF_Backup_Memory_Limit=200000000 (reasonable size: 200 MB)

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**
  - Such as a connection to a RDBMS
- **General: OS-level calls which do not return to the calling Domino code**
  - Network issues (DNSLookup, port problems)
  - Example: AIX filesystem sizeinfo for NFS filesystems (fixed in D6)

How to troubleshoot Server Hangs?

- **Check call-stacks for specific calls**
  - e.g. a large number 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)
  - „Show stat Sem.Timeouts" to check semaphores

- **Run 3 nsd -nomemcheck in short sequence**
  - plus one full NSD

Analyzing Semaphore logs

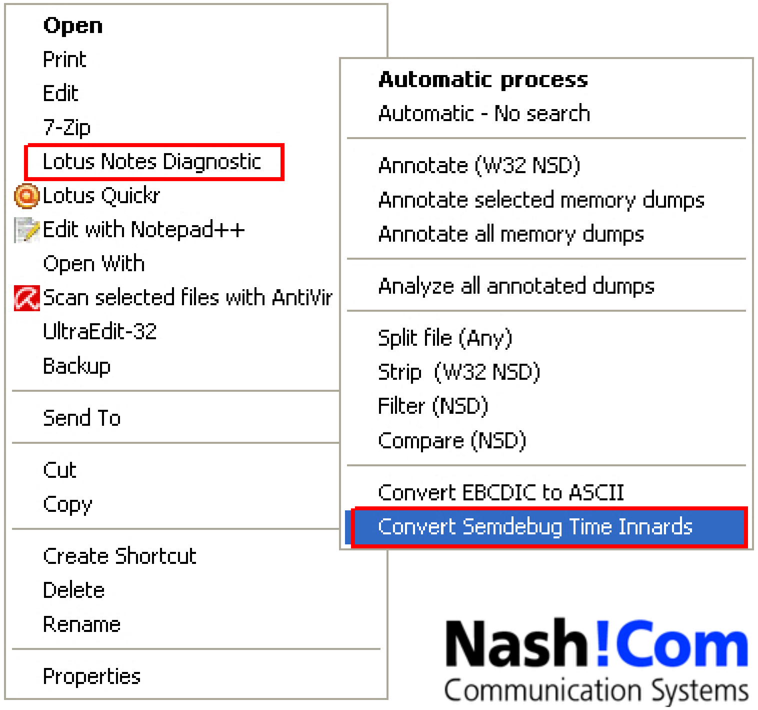
- **semdebug.txt in IBM_TECHNICAL_SUPPORT**
  - contains semaphores locked for more than 30 seconds
  - Information about process/thread, semaphore, time, ...
  - Also contains information who is currently holding the semaphore
    - But just the process/thread.id – You have to annotate on your own via NSD
    - Find 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
```

Annotate Semaphore Logs

- **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



Domino LockManager

- **Lock Manager used when Translog is enabled**
  - Concurrent access to same resources is coordinated and tracked!
- **In case of a hang lock manager could be involved**
  - For example if an important resource (e.g. database) is permanently locked by a process
- **Search for “delay=” in log output**
  - If delay is more than a couple of minutes you usually have an issue
    - Could also occur for single database when compact runs
  - Search for “**Status=Granted”** to see which process/thread is locking the resource

Example Log Held Lock

- **Will be dumped to console.log**
  - Check for “**delay=**” and”**Status=Granted**”

```
LkMgr BEGIN Long Held Lock Dump ------------------
Lock(Mode=X  * LockID(DB DB=F:\appl\calendar\mc_calendar.nsf)) Waiters countNonIntentLocks = 2
countIntentLocks = 0, queuLength = 4
   Req(Status=Granted Mode=X Class=Manual Nest=0 Cnt=2
       Tran=0 Func=N/A dex\stmgr.c:279 [1940:0005-1F38])
rm_lkmgr_cpp:2070
rm_lkmgr_cpp:1306
nsfsem1_c:169
   Req(Status=Waiting Mode=S Class=Manual Nest=0 Cnt=0
       Tran=0 Func=N/A dbopen.c:4073 [1E2C:0002-0F08] Delay=138min)
rm_lkmgr_cpp:2070
rm_lkmgr_cpp:1306
nsfsem1_c:533
   Req(Status=Waiting Mode=S Class=Manual Nest=0 Cnt=0
       Tran=0 Func=N/A dbopen.c:4073 [2518:0002-1D7C] Delay=81min)
rm_lkmgr_cpp:2070
rm_lkmgr_cpp:1306
nsfsem1_c:533
   Req(Status=Waiting Mode=S Class=Manual Nest=0 Cnt=0
       Tran=0 Func=N/A dbopen.c:4073 [1488:0007-021C] Delay=80min)
rm_lkmgr_cpp:2070
rm_lkmgr_cpp:1306
...
LkMgr END Long Held Lock Dump -----------------
```

HTTP Diagnostic

- **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

Debugging Incoming SMTP Messages

- **Generates temp file with full message content before itemization**
  - Message as received by SMTP channel --> Useful for troubleshooting
- **SmtpSaveImportErrors=1**
  - Save if error occurs during message itemization
- **SmtpSaveImportErrors=2**
  - Always save
- **SmtpSaveImportErrors=3**
  - Only save temporary before message conversion and delete after successful conversion. Useful for rare occurring server crashes
- **Extra Tip!**
  - SMTPSaveFileFrom=string in combination with SmtpSaveImportErrors=3
    - keeps log files after conversion if string partially matches with RFC822 “from”
    - Undocumented but very useful to trace issues with certain
    - users or domains in production!

More SMTP Debugging (notes.ini)

- **SmtpSaveOutboundToFile=1**
  - Similar to inbound logging all messages are saved to temporary files
- **SMTPClientDebug=1**
  - Logs RFC821 conversation for outgoing messages
  - Does write to log misc events instead of debug_outfile!
- **SMTPDebugIO=1**
  - Logs transferred bytes
- **SMTPDebugIO=2**
  - Not implemented
- **SMTPDebugIO=3**
  - Logs all RFC822 headers
- **SMTPDebugIO=4**
  - Use this very carefully! Logs also RFC822 data / body!!!

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 per platform

- **You should collect Server stats at least every 15 minutes (default is 90 minutes)**
  - Enable collect task, configure settings in events4.nsf
  - 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 1 partition per OS instance on all platforms

Top Statistics – NSF Buffer Pool

- **Used for Buffering Database I/O**

- **Check Server Stats**
  - Database.Database.BufferPool.Maximum.Megabytes
  - **Database.Database.BufferPool. PercentReadsInBuffer**

- **Interpretation**
  - Bad < 90% < PercentReadsInBuffer < 98% < Perfect

- **Tune: notes.ini NSF_Buffer_Pool_Size_MB=n (in MB)**
  - Default: 512 MB

Top Statistics – NSF Cache

- **Used for Caching Open Databases**
- **Check Server Stats**
  - Database.DbCache.HighWaterMark
  - Database.DbCache.CurrentEntries
  - Database.DbCache.MaxEntries
  - **Database.DbCache.OvercrowdingRejections**
- **Interpretation**
  - Good = HighWaterMark < MaxEntries
  - Good = 0 OvercrowdingRejections
- **Tune: notes.ini NSF_DbCache_MaxEntries = n**
  - Default: NSF_BUFFER Pool size multiplied by 3

Top Statistics – (Cluster) Replication

- **Use to check Cluster Replicator Performance**
- **Check Server Stats**
  - Replica.Cluster.Failed
  - **Replica.Cluster.SecondsOnQueue**
  - **Replica.Cluster.WorkQueueDepth**
- **Interpretation**
  - Perfect < 10 < SecondsOnQueue > 15 > Bad
  - Perfect < 10 < WorkQueueDepth > 15 > Bad
- **Tune:**
  - Add more cluster replicators
  - optimize cluster server usage (e.g. Split active users between cluster mates)

Top Statistics – Transactions

- **Use for Indication of Server Load**
- **Check Server Stats**
  - Server.Trans.PerMinute
- **Interpretation:**
  - Heavy < 30 < Trans.PerMinute (per User) > 10 > Light
- **Tune: Analyze Heavy users and try to avoid load**

Top Statistics – Concurrent Tasks

- **Use to check Simultaneous Active Database Connections**

- **Check Server Stats**
  - Server.ConcurrentTasks
  - Server.ConcurrentTasks.Waiting

- **Interpretation**
  - Waiting should be ZERO

- **Tune:**
  - Server_Pool_Tasks = n (e.g. 80)
  - Server_Max_Concurrent_Trans = m (e. g. Server_Pool_Tasks * Number of Ports)

Top Statistics – Platform Memory

- **Used to check Allocated using memory pools and sub-allocations**
- **Check Server Stats**
  - Mem.Allocated
  - Mem.Allocated.Process
  - Mem.Allocated.Shared
- **Interpretation**
  - Memory Leaks when increasing over days / weeks
- **Tune**
  - By several parameters (bufferpool, cache, namelookup…
- **Note**
  - Be careful interpreting this statistic… Not all memory might be included

Top Statistics – Platform CPU

- **Used to check CPU Utilization on Server**

- **Check Server Stats**
  - Platform.System.PctCombinedCpuUtil
  - Platform.System.PctTotalPrivilegedCpuUtil
  - Platform.System.PctTotalUserCpuUtil

- **Interpretation:**
  - OK < 90% CombinedCpuUtil > 90% > TOO HIGH
- **Tune**
  - Many Root Causes Possible

Top Statistics – Paging File

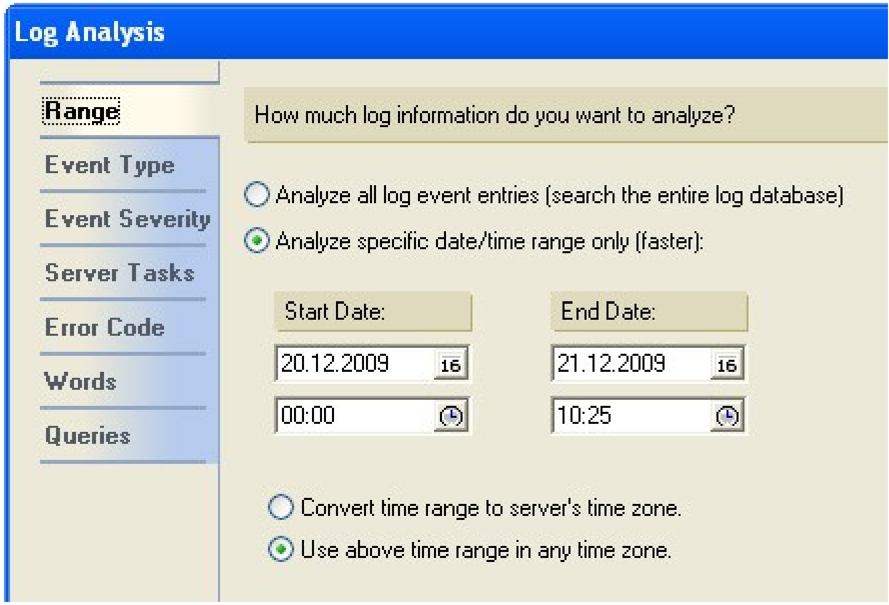
- **Use to check Server Memory Swapping to Disk**
- **Check Server Stats**
  - Platform.PagingFile.Total.PctUtil
- **Interpretation**
  - OK < 0% < PctUtil.Avg > 10% > BAD
- **Tune**
  - OS Level tuning, Check Memory

Top Statistics – Platform Disk

- **Used to check Disk Performance**
- **Check Server Stats**
  - Platform.LogicalDisk.1.AvgQueueLen
  - Platform.LogicalDisk.1.PctUtil
- **Interpretation**
  - Good < 2% < AvgQueueLen > 5% > BAD
  - Good = PctUtil < 80%
- **Tune**
  - By several parameters (bufferpool, cache, namelookup
- **Note**
  - Platform.LogicalDisk.1.AssignedName=C points to the disk

Analysis Tools

- **Domino Admin Client contains analysis Tools**
  - Located in Server/Analysis Tab
  - Cluster Analysis
  - Log Analysis
- **You should regularly analyze server logs**
- **Activity logging can also help for troubleshooting**
  - Needs to be enabled in Server Config Document



How much log information do you want to analyze?

○ Analyze all log event entries (search the entire log database)

● Analyze specific date/time range only (faster):

Start Date:	End Date:
20.12.2009	21.12.2009
00:00	10:25

○ Convert time range to server's time zone.

● Use above time range in any time zone.

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

Nash!Com
Communication Systems

Still an Issue: Broken Design Collection

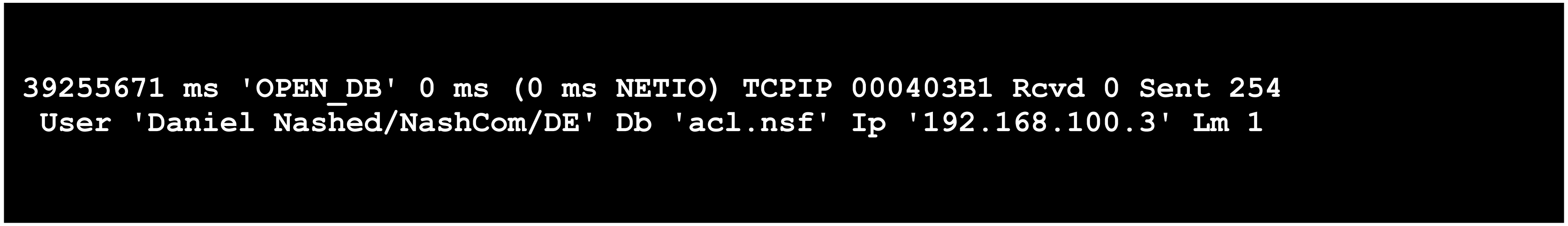
- **Domino has an internal design cache in each database to find design notes**
  - Used by NIFFindDesignNoteExt (Transaction: FINDDESIGN_NOTES)
- **In some odd cases the design cache breaks**
  - Without the design cache the client tries to find design elements the “old style” by opening and searching 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 or hotfix (also only work-around)**
  - Reference: SPR #RSTN7K2EM4, TN #1322578 Performance degradation using "Private on First Use" views or folders

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**
  - Has first been implemented thru hotfixes for SAI troubleshooting and finally helped fixing SAI :-)

Additional Server_Clock since 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

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
  - Notes/Domino also has many features to troubleshoot performance issues on client and server side

- **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 ...

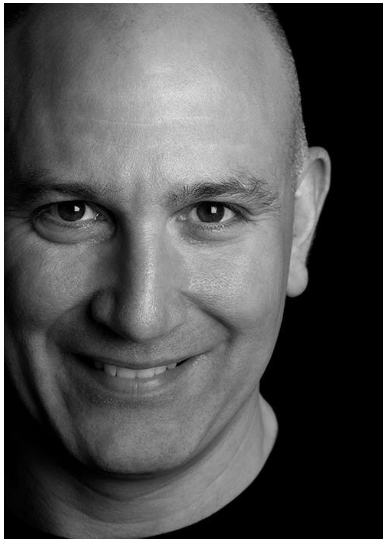
Resources

- **Free OpenNTF Client_Clock Parser**
  - http://www.openntf.org/Projects/pmt.nsf/ProjectLookup/Notes%20RPC%20Parser
  - Gives you a nice way for parsing client_clock
  - But does currently not annotate design element names etc

- **Crash program for client and server from IBM**
  - Not longer available
  - If you really need one for testing, ask me for “nshcrash”

Q&A

- I hope you enjoyed the presentation
- Questions now or later?
- Please, fill out your evaluations!
- Contact
  - nsh@nashcom.de
  - http://www.nashcom.de
  - http://blog.nashcom.de



IBM CHAMPION

Nash!Com
Communication Systems