Domino Compact Update

Düsseldorf, 24 November 2016

Daniel Nashed, Nash!Com

Domino 9 Compact – New Options

- **-# nn**

  - Execute up to nn compactions in parallel using different threads.
    Up to a maximum of 20 Threads with a default of 1.

- **-W nn**

  - Only compact databases which haven't been compacted successfully in last nn days.
    Ordered by last compact time.

- **-w**

  - Exclude system databases (e.g. log.nsf, names.nsf).

- **-X nn**

  - Limit each database compaction to no more than nn minutes.
    (applies to in-place compaction only)

- **-x nn**

  - Limit total compaction time to no more than nn minutes.

- **-***

  - Include *.ntf and *.box when processing a directory

Upgrading ODS for DB with old DB Class

- **If you used explicit file versions for a database like test.ns7 you cannot use compact to bring the database to a newer ODS level**

  - Even when you change the extension of the database, the internal database header has a fixed version which will prevent the ODS change

- **"-upgrade Upgrade databases created with older DB classes to the most recent class."**

  - For example: **compact -C -upgrade help/** will change the ODS for all help databases

  - Tip: **-ODS** Perform a copy-style compact only if the current ODS is less than desired default ODS.

- **Extract from C-API header file**

  - **#define   DBCLASS_NOTEFILE   0xff01**

  - #define   DBCLASS_V6NOTEFILE   0xff10

  - #define   DBCLASS_V8NOTEFILE   0xff11

  - #define   DBCLASS_V85NOTEFILE   0xff12

Nash!Com
Communication Systems

DNUG

Database Management Tool - DBMT

- **New Multi-threaded Domino Server Add-In Task**

- **Performs automated coordinated administrative tasks on databases**

- **Standard Maintenance**

  - Copy-style Compacting (compact) → Only copy-style compact!!

  - Fixup of Corruption (fixup)

  - Full Text Indexing (updall)

  - Incremental View Indexing (update)

- **Additional Maintenance**

  - Folder Reorganizing

  - Deletion Stub Purging

  - Expiring of Soft Deleted Entries

  - Updating of Unread Lists

Database Management Tool - DBMT

- **DBMT replaces the need for running Compact on Non-System databases**

  - Revisit your current compact strategy

- **DBMT replaces the need for running of Updall**

  - If DBMT scheduled daily, suggest removing Updall from ServerTasksAt2= in notes.ini
  - Tip: SetupLeaveServerTasks=1 so any future upgrade will not put it back

- **DBMT eliminates awkward long pauses that may occur for in-line operations**

  - Database is opened for first time e.g. "Releasing unused storage for...."

- **Multi-Threaded Servertask**

  - Options from 1 to 100 Threads
  - Usually not more than 10 threads for all operations make sense

Nash!Com
Communication Systems

DNUG

DBMT command line options

- **-compactThreads <n>**

  – Use <n> threads for compact where **n** is between **0 & 100** inclusive

  – Number of threads DB copy style compact operations. **Default is 1 thread**.
  If **0** is specified, no compact operations are performed.

- **-updallThreads <m>**

  – Use <m> threads for updall, where m is between **0 & 100** inclusive

  – Number of threads doing updall operations. Default is **1**.

- **-ftiThreads <g>**

  – Use <g> threads for the rebuild of the full text indices, where g is between **1 & 100** inclusive

  – Configures number of threads for rebuild of full text indices if corrupt or **-ftiNDays** specified. Default is **1**; 0 is not allowed.

- **-ftiNdays <f>**

  – **Rebuild** full text indices every <f> days from initial creation

  – Default is to rebuild full text indexes only when found corrupt.

Nash!Com
Communication Systems

DNUG

DBMT Schedule Options

- **-range <s> <e>**

    - only run between times <s> and <e> daily

    - Minimum of 10 minutes between time <s> and time <e>

    - Times must be specified in 12hour format with a required AM/PM (e.g. 11:50PM)

    - DBMT process will remain loaded but dormant until -range comes around again the next day

- **-timeLimit <q>**

    - only run for <q> minutes

    - DBMT process will exit once **-timeLimit** expires and all active DBMT processing completes

    - -stoptime <e>

    - only run between now and time <e> once

    - Time must be specified in 12hour format with a required AM/PM (e.g. 6:00AM)

    - DBMT process will exit once -stoptime is reached and all active DBMT processing completes

DBMT Compact Options

- **-compactNdays <p>**

  - Compact databases that have not been compacted in last <p> days

- **-noCompactLimit**

  - Allow compact to run past the end time to finish, but no new compacts will be started

- **-force <n>**

  - Take DBs offline for fixup & compact if flagged for force
    - Fixup is run only when 5 or more consecutive compact operations fail (and the failure is not due to database in use).
  - <n> between 0 and 7 where 1= Sunday, 2 = Monday, and so on.
    - If n is 0 (zero), the fixup operation will run any day.

Nash!Com
Communication Systems

DNUG

DBMT for System Databases

- **DBMT does not compact System Dbs by default**

  - but will perform -updall and -fti operations

  - You can explicitly specify databases on command-line or "ind" file

- **System-Database List**

  - names.nsf

  - events4.nsf , admin4.nsf, catalog.nsf, ddm.nsf

  - log.nsf, statrep.nsf, domlog.nsf, lndfr.nsf

  - busytime.nsf, cldbdir.nsf, clubusy.nsf

  - daoscat.nsf, dbdirman.nsf, dircat.nsf

  - mtdata\mtstore.nsf

DBMT Compact Filter indirect file

- **dbmt_compact_filter.ind**

    - Exclude list for Databases

    - DBMT will not compact database(s) **< ODS41** – Database(s) automatically added to dbmt_compact_filter.ind by DBMT

    - If -range is specified and **-noCompactLimit** is not specified

        - If DBMT compact of database takes longer than ~full -range specified and terminated because end of range reached , database automatically added to dbmt_compact_filter.ind by DBMT

        - Prevents DBMT from churning on same database every night

    - Administrators may manually add additional database(s) to **dbmt_compact_filter.ind** if e.g. they have an Application/database that needs to be Highly Available all the time

DBMT -updallThreads

- **By default, specific views in mail databases are**
  - brought up-to-date
  - **Created**
  - **marked non-discardable**
  - dramatically improving end user's fail over experience in a cluster

- **Need Inheritance Enabled and Template Name one of StdR7Mail,StdR8Mail,StdR85Mail or StdR9Mail**
  - ($Inbox), ($Drafts), ($Sent), ($All)
  - ($RepeatLookup), ($ToDo), ($Calendar), (Haiku_TOC)
  - ($Alarms), (iNotes), ($Users), (iNotes_Contacts), ($ThreadsEmbed)

- **If you use customized mail template(s) use the following notes.ini**
  - DBMT_MailTemplate=templatename1,templatename2,templatename3

DBMT -updallThreads

- **Custom Database View Configuration**

  - Required Inheritance Enabled and Template Name set on database(s)

  - DBMT_TemplateName=ViewNameOrAlias1;ViewNameOrAlias2;...ViewNameOrAliasN

  - Substitute TemplateName after the DBMT_ with inherited database Template Name

  - Substitute the ViewNameOrAlias1-N with database View Name or Alias and separate the ViewNameOrAlias1-N with either semicolons or commas

- **Examples in notes.ini**

  - DBMT_StdNotesLog=SecurityEvents, ReplicationEvents, MailRoutingEvents

  - DBMT_StdR4AdminRequests=All Requests by Time Initiated;All Request by Server

- **Remember notes.ini setting(s) are restricted to 128 characters!**

DBMT Statistics

- **Stastistics for Compact and Updall when DMBT terminates**

  - DBMT.Compact.Began 04.06.2013 21:27:44

  - DBMT.Compact.Finished 04.06.2013 21:27:59

  - DBMT.Compact.Successful 26

  - DBMT.Compact.Unsuccessful 1

  - DBMT.Compact.Unsuccessful.InUse 1

  - DBMT.Compact.Unsuccessful.TimeLimit 0

  - DBMT.Compact.BackLog 0

  - DBMT.Compact.Fixup 0

  - DBMT.Updall.Began 04.06.2013 21:27:44

  - DBMT.Updall.Finished 04.06.2013 21:28:03

  - DBMT.Updall.Processed 36

  - 04.06.2013 21:28:05   Database compactor process shutdown

Multiple DBMT Processes

- **DBMT takes as input for database(s) to act on:**

  - A single database (.nsf)

  - An indirect file (.ind)

  - A directory name

  - No specification - Runs against all databases

- **Restricted in order to prevent overlap**

  - <u>Only one</u> DBMT process running against <u>all databases</u> is allowed to be loaded at any given time.

    - Any additional DBMT process launched to run against all databases will not be allowed and will simply exit

  - Administrators are allowed to run multiple DBMT processes by specifying a database, indirect file or directory name, but consideration should be given before doing so to avoid overlap

  - DBMT by default, does not compact system databases, unless the system database (.nsf) is specified explicitly on the command line or explicitly in an indirect file (.ind)

Domino 9 Updall

- **Updall performs the following tasks by default.
  These are also tasks that the database maintenance tools
  performs:**

  - purges deletion stubs

  - expires soft deleted entries

  - updates unread lists

- **New -nodbmt option**

  - Skips those operations for updall if you have DBMT configured

  - Faster if you run updall for other reasons

DBMT Limitations

- **For Compact Options you cannot specify how much Space should be free for compact to run (-S parameter)**

  – But you can configure how often a database should be compacted

- **Compact only runs in copy-style compact**

  – Works fine in a cluster

  – Else you need to make sure compact only runs on weekends/ middle of the night

- **You should only have one DMBT with -range Option**

  – No second DMBT for the weekend with a bigger time window

  – Means you should keep compact operations separate and run them during the day on weekends

    - In case you have archive style transaction logging you need to schedule compacts for the weekend anyway

    - Tip: Ensure you have the right quota setting in the Translog tab:

      – "**Check space used in file when adding a note**"

DBMT Examples

- **dbmt -compactThreads 4 -updallThreads 4 -ftiThreads 4 -ftiNdays 30 -range 1:00AM 6:00AM -compactNdays 7 -force 1**

  - Compact all databases not compacted for 7 days with 4 threads

  - Update all views with 4 threads

  - Update FT Index with 4 threads, Rebuild if corrupt or older than 30 days

  - Run between 1 AM and 6 AM every night and stay in memory during the day

  - Fixup database which did not compact sucessfully 5 times on Sunday

D9.0.1 - DBMT – File Space Pre-Allocation

- **A copy style compact generates a temp file and copies the database note by note into the database**

  - Allocation of file-space is also done step by step

  - Depending on the file-system, free space etc the database is already fragmented when the compact finishes – it is very likely that the database has many fragments

- **We have been asking for this option for many years**

  - Currently only implemented for DBMT but on the wishlist for standard compact and C-API

- **In D9.0.1 DBMT will pre-allocate the calculated space that the physical database needs (it's DAOS aware)**

  - Enabled by default in 9.0.1

  - Notes.ini **DBMT_PREFORMAT_PERCENT=n** allows you to increase or decrease the size

  - For example DBMT_PREFORMAT_PERCENT=120 will have 20% contiguous free space in the database after compact (if there is sufficient unfragmented space in the file-system)

Nash!Com
Communication Systems

DNUG

Mail File Disable Compact Abort

- **Since Domino 8.5.3FP3**

- **By default, compact operation will be aborted when mail is delivered**

  - To make DBMT more efficient set notes.ini **MailFileDisableCompactAbort=1**

  - New mail in mail.box will sit undelivered in mail.box on the Domino Server

  - Mail router will periodically retry delivering new until the current compact completes

- **Especially for large mail files, this may be an unacceptable extended period of time to go without receiving new email**

  - But there is another new option in Domino 9.0 (see next slide)

- **Best practice is to set**

  - MailFileDisableCompactAbort=1

Nash!Com
Communication Systems

DNUG

Domino 9.0
Mail File Enable Delivery Failover

- **Since Domino 9.0,mail Router has ability to failover mail delivery to an Available mail file replica on a cluster mate when the local replica is unavailable because:**

  - Copy-style compact of mail file is in progress

  - Fixup of mail file is in progress (Performing consistency check on...)

  - Mail file is missing (File does not exist)

  - To enable delivery failover, set **MailFileEnableDeliveryFailover=1** in notes.ini on Domino Server

  - **DBMT** uses the replica index in cldbdir.nsf to determine if it should compact the mail file on this Cluster Mate on this day to make sure all Cluster Mates are not compacting same replica on same day

    - **Only works in combination with -compactNdays**

  - **$MailClusterFailover** item will be appended to the note with a value of the Domino Server name of the server where the note was actually delivered on

  - Note: For Mail-Rules you need to allow your servers to use "monitors" (Security tab)

Domino 9.0 Compact Replication

- **Should only be used in situations where its known that the ID table becomes fragmented.**

  - The goal is ID table reorganization and maintaining Views and Unreads

  - There is additional overhead resulting in slower performance

  - Hence, compact replication is not a replacement for all in-place or copy-style compaction

- **Compaction that creates a replica of a database under the covers of compaction while the source database remains accessible**

  - Provides the benefit of internally re-organizing the IDs in the new replica database to avoid the ID table fragmentation that may eventually result in access failure

  - Extremely useful for preventative maintenance to avoid fragmentation and recovery of a database that is highly subject to becoming inaccessible due to fragmentation

- **Maintains View and Unreads between the old and new replica**

Nash!Com
Communication Systems

DNUG

Compact Replication command line options

- **-REPLICA**

  - (Required) Compact via the background replication mechanism

- **-REN_WAIT=\<nn\>**

  - (Optional) length of time in \<nn\> minutes for compact process to continue to try and rename after the replication and synchronization phase is complete

- **-RESTART**

  - (Optional) specify if a server restart should occur if the rename phase is not successful to allow the rename phase to complete on Domino server startup

- **-IDS_FULL=\<nn\>**

  - (Optional) Only initiate compact replica if IDTable fullness estimate >= \<nn\> percent
  - 0 means all the time

- **Example: compact mail.nsf -REPLICA -IDS_FULL=80 -REN_WAIT=3 -RESTART**

Compact Replication – How it works

- 1) Optional IDTable analysis - Calculate fullness of ID table to determine need for compaction

- 2) Creates and synchronizes temp **.REPL** database with original **.NSF** database - Get initial full set of notes

  - a) Note copy phase - Outputs to temp **.REPL** database the notes, keeping existing and deleted notes in contiguous sets to avoid ID table fragmentation

  - b) View synchronization - Build views in temp **.REPL** database based on views from original **.NSF** database

  - c) Unread synchronization - Build unreads in temp **.REPL** database based on unreads from original **.NSF** database

- 3) Replace

  - a) Drop all users from original **.NSF** database

  - b) Take original **.NSF** database offline

  - c) Rename original **.NSF** database to **.ORIG** database – Rename new temp **.REPL** database to new **.NSF** database

- 4) Synchronize new **.NSF** database with old original **.ORIG** database - Get any new notes in **.ORIG** since initial sync in 2)

  - a) Note copy phase of new notes - outputs to **.NSF** database any new notes, keeping existing and deleted notes in contiguous sets to avoid ID table fragmentation

  - b) View synchronization – Refreshes views in **.NSF** database based on views from old original **.ORIG** database

  - c) Unread synchronization - Refreshes unreads in **.NSF** database based on unreads from old original **.ORIG** database

- 5) Bring new **.NSF** database back online.

- 6) If phase 3) does not complete due to a soft time out (not errors), a Domino Server restart can be applied to complete the operation as the Domino Server restarts at the initial open of the **.NSF**

Nash!Com
Communication Systems

DNUG

Compact Replication Example

- **Additional Benefit for databases always in use such as system databases**

    - system.ind is indirect file listing system databases log.nsf, names.nsf and admin4.nsf

    - compact system.ind -replica -restart

    - Administrator requests a compact replication of the 3 system databases listed in the indirect file system.ind.

    - Since these are system files and always in use, a Domino Server restart is specified

    - Restart will be initiated once ALL system database compact replications are ready for the rename phase

    - This can reduce Domino Server down time in a maintenance window where Administrators would quit Domino Server, Copy Style compact the 3 system databases (even concurrently) and restart the Domino Server

Nash!Com
Communication Systems

DNUG

Programm Document Change

- **Already in since 8.5.3 FP3**

- **Supported values for Program record "Server to Run on" field:**

  – Full hierarchical name of a server to run on (e.g. Sales01/Renovations)

  – A "**Cluster Name**" to run on all servers in the Cluster

  – **An asterisk** in leftmost component of the hierarchical name to run on matched servers (e. g. */Renovations) An asterisk to run on all servers (*)

    - "*" worked before but is now officially supported

Domino 9.0
GroupName + Pattern in Program Record

- **New in Domino 9.0**

- **You can use a "Group Name" where the group contains the list of servers to run on**

  - Group type: must be "<u>Servers only</u>" (not a multi-purpose group)

- **A pattern-matching question mark (?) character**

  - The question mark (?) -- allows you to include all servers where one or more subsequent characters in the server name vary

    - Example: Sales??/Renovations

      - Includes Sales01/Renovations, Sales02/Renovations, etc

- **An asterisk (*) anywhere within a name**

  - no longer limited to the leftmost component of the hierarchical name

    - Example: Sales*/Renovations

      - Includes Sales001/Renovations, Sales002/Renovations, and so on

Nash!Com
Communication Systems

DNUG

Q&A

- **Thanks for your attention!**

- **Contact**

  - http://blog.nashcom.de

  - http://www.nashcom.de

  - nsh@nashcom.de

  - +49 172 2141912

Nash!Com
Communication Systems

DNUG