# Keep your Domino Servers Humming – The Wonderful World of Domino Server Maintenance

Cormac McCarthy – Domino People Ltd

Engage 2026 – Ghent

# Cormac McCarthy – Domino People Ltd

Domino Admin – 20+ years experience

Experience across a wide variety of areas on the Domino platform including Upgrades, Mail, Migrations, Audits, Performance, Integration, Consolidations, Security, Project Management and everything Infrastructure.

I write a blog at dominopeople.ie/blog

Work for a HCL Business Partner based in Ireland – Domino People Ltd.

HCL Ambassador for the last six years from 2021-2026




Email: cmccarthy@dominopeople.ie
Linkedin: https://www.linkedin.com/in/cormacmccarthydomino
Or come up to me and say hello this week.

# Who is this Presentation for?

Domino Admins.  (though Developers might get something from it as well).

# What will presented today?

The fundamentals of maintenance (i.e. the actual maintenance tasks in Domino and what are they for).

Best Practices on preventative maintenance (and some tips on reactive maintenance) on Domino Servers.

Key questions Domino Admins should be asking themselves on an ongoing basis on how best to manage and maintain their infrastructure.

What is Maintenance? The act of keeping something in good condition by checking or repairing it regularly.

The goal is that we will keep Domino Servers humming – (i.e. Online and functioning as they should)

There is a tonne of material to get through here.  I will speak too quickly.  The slides will be available afterwards.

If you makes changes to your environment (or day to day practices) please test any changes before rolling out.

# #1 Configuration principles to abide by

Keep your Domino Server on a supported version of Domino on a supported version of the OS Host. (keeping internet facing services on latest release).

Do not keep any debug running on your servers for anything other than short term.

Keep your databases on the latest version of ODS (from Domino 14.5 it's easier to work with system databases as there is a dominosystemdbs.ind file). From 12.0.2 you don't need Create_r*_databases=1 anymore just a copy style compact or dbmt -ods)

Turn on transaction logging by default.

You probably have too many connection documents. As long as name resolution works. You don't need connection documents for "Notes mail" in the same Notes Named Network (i.e. NRPC traffic). For replication – if you have clusters (again as long as name resolution works) as long as the local cluster.ncf is correct, you should only need to connect to the cluster name in a connection document

# #1 Configuration principles to abide by

- Configuration documents don't have to be per server, you can apply settings to all servers and sets of servers (groups) If you have clustered servers or servers that provide a similar function (i.e. SMTP, HTTP Servers) it can make maintaining consistency more straightforward to apply settings to a group of servers.

Your app servers very likely don't need to be running Calconn, RnRMgr and Sched.  (check servertasks= post upgrade)

Recreate your log.nsf files often, rename the old file with the server offline and a new one will be created as long as log.ntf exists (anytime you take the server offline for an upgrade etc)

# #2 DBMT

**DBMT** is a flexible, controllable, scalable version of updall, with options to compact and fixup.

My experience is that the “average” Admin have been relatively slow to adopt. Maybe as there are so many possible switches it's can be intimidating to pick up and start using if you've no experience of it. Also is doesn’t have exact feature parity with the updall, compact, or fixup.

Massive advantages over traditional maintenance tasks– it allows multiple threads for running an updall and compact, allows stop and start of maintenance and allows efficient proactive use of fixup (i.e. run a fixup if compact continues to fail), it also has a tonne of switches the other tasks don’t have. It is by far the most scalable maintenance task.

Quick note to set MailFileDisableCompactAbort=1 if you’re using Domino based mailfiles. It will stop mail delivery interrupting compacts.

# #2 DBMT

Example Usage:

*Load dbmt -compactThreads 8 -updallThreads 8 -range 2:00AM 7:45AM -compactNdays 5 -force 1*

This set of parameters:

- -Specifies 8 separate threads for both the compact and updall tasks
- -Only runs in a window between 2:00 AM and 7:45 AM
- -Compacts non-system databases every 5th day
- -On Sunday performs a fixup on databases that cannot be compacted.

You shouldn't be in doubt about what happened when DBMT runs as it gives excellent output to the console, e.g.

```
DBMT.Compact.Began          23/01/2026 02:00:21
DBMT.Compact.Finished       23/01/2026 04:08:29
DBMT.Compact.Successful             381
DBMT.Compact.Unsuccessful           2
DBMT.Compact.Unsuccessful.InUse             0
DBMT.Compact.Unsuccessful.TimeLimit         0
DBMT.Compact.BackLog        2
DBMT.Compact.Fixup          0
DBMT.Updall.Began           23/01/2026 02:00:21
DBMT.Updall.Finished        23/01/2026 02:03:04
DBMT.Updall.Processed       395
```

# #2 DBMT

Once you've tested a set of parameters (and refined as needed) you can run in a program document at server startup if you're using the range switch (rather than traditional maintenance tasks that are run at specific times and have no "memory" of previous runs).

| Basics | | Schedule | |
|---|---|---|---|
| Program name: | dbmt | Enabled/disabled: | At server startup only |
| Command line: | compactThreads -8 updallThreads 8 -range 02:00AM 07:45AM -compactNdays 5 -force 1 | | |

After restart you should see that the DBMT Manager task is idle with the switches you've specified.  If not you'll likely see a syntax error in the log.

| | |
|---|---|
| DBMT Manager | Idle -compactThreads 8 -updallThreads 8 -range 2:00AM 7:45AM -compactNdays 5 -force 1 |

There's a danger sometimes when introducing DBMT that people get overwhelmed by the number of options, but you should be aware that there are new switches available for items like:

performing dbmt on system databases (-systemDBs or -sd)
regular expressions (-regex or -re changes in 14.5.1 as per
https://help.hcl-software.com/domino/14.5.1/admin/admn_general_expressions_dbmt_tool.html)

For more info
https://help.hcl-software.com/domino/14.5.1/admin/admn_running_the_database_maintenance_tool_from_a_program_document_t.html

# #2 DBMT

DBMT also purges deletion stubs, expires soft-deleted entries, reorganizes folders, and merges full-text indexes.

The compact style compact it uses is more effective than running explicitly via compact as it reallocates the size of the .tmp file when doing the compact.  This reduces disk fragmentation.

Some Notes:

> HCL explicitly state that you shouldn't use scheduled Updall if you are using DBMT. (i.e. remove it from servertasksat2= in your notes.ini)
>
> The update task does run on your servers during the day, how efficiently it deals with views and indexes is dealt with elsewhere in the session.
>
> Running DBMT can be highly intensive. i.e. run out of hours.
>
> Don't set up DBMT on all your servers without assessing impact of same.

My use case for DBMT in 2026?
Should be the default program document you use to scalably apply updall and compact to a large number of databases.

# #3 Updall

**Updall** is a maintenance task in Domino used to explicitly force an update to views and indexes in databases.

It can be run against single databases, a list of databases, .ind files or directories.

There are different options for running with switches (used views, unused views, specific views, indexes only etc) - https://support.hcl-software.com/csm?id=kb_article&sysparm_article=KB0037060

My use case for updall in 2026?

- For targeted troubleshooting on one or a small number of databases
- When a new (large) replica of a is created and you want the views to be accessible quickly. (updall -C)
- When specific views in names.nsf (or other system databases) need to be updated immediately for services to function properly.
- Ultimately with DBMT you don't NEED to use updall any more BUT in my opinion there is no harm in using it in once off cases. It just isn't as scalable as DBMT.

https://support.hcl-software.com/csm?id=kb_article&sysparm_article=KB0037060

# #4 Compact

Compact - generally either runs to remove empty space from a database or to clean corrupt or unwanted structures from a database.

My use case in 2026- can be use for targeting options on individual databases - where you want to use the following switches that cannot be run with dbmt, also where you don't want to run updall

-i -c (this should be a last resort)
-a (archive)
-REPLICA (creates a new replica in the background while compacting where the original database is available during the rename)
-c daos on/off
-c nifnsf on/off
-t (transaction logging on)
-T (transaction logging off)
-v (Data compression on)
-V (Data compression off)
and more niche switches here - https://help.hcl-software.com/domino/12.0.2/admin/tune_compactoptions_r.html

# #5 Fixup

Fixup is generally an emergency only task.

It runs automatically if database(s) are held open when a server crashes/hangs (i.e. “consistency checks”).

Run in the case if there is a suspected data corruption.  i.e. if you cannot open a database - or are seeing errors like "B Structure invalid"

Switches:
for transaction logs -J

Also I’d generally advice using -F switch (scan all documents as opposed to just scan documents not scanned since last fixup).

https://help.hcl-software.com/domino/14.5.0/admin/admn_fixupoptions_r.html?hl=fixup

# #6 Offline versus Online Maintenance

Online maintenance simply means the Domino server is up while completing maintenance (either scheduled or by exception) It is usually the standard way of running maintenance tasks.

However there are exceptions to this that should be considered namely:-

ODS- upgrades

Also, if you have ongoing repeated issues with replicas on a particular server such as:

- compacts not reducing sizes of databases adequately
- compacts not running as users are active
- compacts not completing because agents are running
- views not updating quickly enough overnight
- database corruption causing (repeated) server crashes
- etc

Offline maintenance may be a good option for these use cases.

Offline maintenance is when the Domino server/service is offline but the OS/Host is still up while completing maintenance from a command prompt/terminal. More here - https://support.hcl-software.com/csm?id=kb_article&sysparm_article=KB0096018

# #6 Offline versus Online Maintenance

The main advantage of offline maintenance is that the maintenance is not interfered with by anything else, which makes it generally more effective than online. All server resources should be allocated to completing the job at hand.

Can be run with exactly the same switches as if the server is online. Just take the server offline and run the task manually.

One subtle significant difference between pre Domino 14 offline maintenance in Windows and post Domino 14. As the notes.ini has moved to the data directory you need to run maintenance tasks from the data directory.

Windows example (Pre-Domino 14):

```
c:\users\Administrator\>cd c:\HCL\Domino\

c:\HCL\Domino>nfixup.exe -F -J d:\HCL\Domino\Data\dodgydb.nsf
```

Windows example (Domino 14+):

```
c:\users\Administrator\>cd c:\HCL\Domino\data

d:\HCL\Domino\data>C:\HCL\Domino\nfixup.exe -F -J dodgydb.nsf
```

# #6 Offline versus Online Maintenance

Linux example:

```
/opt/hcl/domino/bin/fixup -F -J /local/notesdata/dodgydb.nsf
```

iSeries example:

```
RUNDOMCMD SERVER(DOMINO SERVER NAME) CMD(CALL PGM(QNOTES/fixup)
PARM('DODGYDB.NSF' '-F' '-J')) BATCH(*YES)
```

# #7 Cluster Management

Symmetrical clusters - R.10 feature - still not universally applied
https://help.hcl-software.com/domino/12.0.2/admin/sym_cluster.html

Smart, configurable set of tasks that runs on every node in a cluster. They have the potential to detect and repair damaged databases or missing databases, and .nlo objects within a cluster.

Some restrictions to it's usage but in general: if you apply settings "consistently" within your cluster (DAOS, ODS, and you don't have directory links or applications in the root, and you have enough disk space), then you can use it.

If you haven't already applied it apply it iteratively (don't turn everything on immediately).

# #7 Cluster Management


To configure what should be repaired it's the Basics tab of the cluster configuration document.

It's quite configurable i.e. you can set:
Folders to maintain, whether automatic repair fixes or reports (or does nothing), the scan interval, and what to do with damaged files.

To actually turn on it's in the notes.ini - *D10_ENABLE_REPAIR=1*
and load AutoRepair and RprCleanup tasks (add to ServerTasks= Line in server notes.ini)

There are multiple query options to do with the repair command as per here - https://help.hcl-software.com/domino/14.5.0/admin/sym_cluster_repair.html

There are also many tuning options as per here - https://help.hcl-software.com/domino/14.5.0/admin/sym_cluster_tune.html

# #8 Database Sizing

Limits on .nsf file prior to R.10 ODS was 64 GBs, since it's 256 GBs

Increasing the limit is "good" overall BUT there can be more to maintaining large databases than there would have been with smaller.

In terms of maintaining a database the following is a reasonable list of questions that an Admin should be asking themselves when considering large databases (I cover many of these further separately in the presentation):

How does the database size affect your backups?

Do regular (nightly) maintenance tasks run? (without problems)

# #8 Database Sizing

Do agents run without error? What are the runtimes of your agents?  Do scheduled agents run when they are meant to?

What are the document counts on the database(s)?

How many all document views are there? (Are they all required and how many have sortable columns).

What are the size of the views (right click in the admin client – manage views)? (both in total and as a percentage of the overall database)? Do you need to look at moving the views out of the database?

What is the user perception of the performance of the database(s)?

# #8 Database Sizing

What type of FTIs do you have?

Are there regulatory requirements? (for example GDPR – should you be retaining old information).

How much data is actually active? Can you archive old information to another database? (Split into years is a good practice, not necessarily all data should only be 1 year old, but pick a point in time in which to create yearly archives).

What is the space used of the database(s) – is  i.e has compact being working effectively on it/them.

# #9 Domino Disks

There are multiple things to consider with Domino on disks.  Much of this relates to initial setup or performance rather than maintenance but we'll focus on some core concepts.

How much free disk space on your data drive do you need to maintain Domino?

I think it's something like: (The amount the data grows YoY)

+(the size(s) of the largest database(s) you intend to run copy style compacts on concurrently)

+P(Unused views being built)*(the size of those unused views)

+(10-15 %)

My message here isn't allocate too much free space "just in case".  It's that calculating the amount of free space you need can be complex. Be sensible and pragmatic and make sure your servers will operate, based on above.

Reminder – if you are running Domino as an Enterprise application you shouldn't be "getting by" with minimal amounts of free space.

# #9 Domino Disks

A second consideration around disks is primarily one around metrics of performance.

MAX IOPS and burst speeds are commonly quoted on cloud (particularly M365 cloud), but one item to make sure is measured properly is crucial – disk latency.  The actual time time taken to carry out a single input/output on disk.

Be as prescriptive as possible on what are your requirements for each disk (this takes experience and knowledge of the systems).

For example in Windows prescribe your C disk needs to be able to perform with a disk latency of under 3ms.

# #9 Domino Disks

You can test it yourself with diskspd.
https://learn.microsoft.com/en-us/azure/azure-local/manage/diskspd-overview?view=azloc-24112

Example syntax :

*diskspd -c10m -d30 -r -w40 -t4 -o32 -b4k -Sh -L c:\diskspd\testfile.dat*

Here's a sample of what this gives you for each disk:

```
Total IO
thread |       bytes     |     I/Os     |     MiB/s   |  I/O per s |  AvgLat  | LatStdDev |  file
-----------------------------------------------------------------------------------------------------
     0 |       246190080 |        60105 |       7.83 |    2003.22 |   15.968 |    20.776 | c:\DiskSpd\testfile.dat (10MiB)
     1 |       219525120 |        53595 |       6.98 |    1786.25 |   17.907 |    21.531 | c:\DiskSpd\testfile.dat (10MiB)
     2 |       247468032 |        60417 |       7.87 |    2013.62 |   15.885 |    20.759 | c:\DiskSpd\testfile.dat (10MiB)
     3 |       248844288 |        60753 |       7.91 |    2024.82 |   15.798 |    20.750 | c:\DiskSpd\testfile.dat (10MiB)
-----------------------------------------------------------------------------------------------------
total:         962027520 |       234870 |      30.58 |    7827.92 |   16.345 |    20.957

Read IO
thread |       bytes     |     I/Os     |     MiB/s   |  I/O per s |  AvgLat  | LatStdDev |  file
-----------------------------------------------------------------------------------------------------
     0 |       147939328 |        36118 |       4.70 |    1203.77 |   14.734 |    20.991 | c:\DiskSpd\testfile.dat (10MiB)
     1 |       131907584 |        32204 |       4.19 |    1073.32 |   16.477 |    21.653 | c:\DiskSpd\testfile.dat (10MiB)
     2 |       148152320 |        36170 |       4.71 |    1205.50 |   14.477 |    20.888 | c:\DiskSpd\testfile.dat (10MiB)
     3 |       149741568 |        36558 |       4.76 |    1218.43 |   14.406 |    20.876 | c:\DiskSpd\testfile.dat (10MiB)
-----------------------------------------------------------------------------------------------------
total:         577740800 |       141050 |      18.36 |    4701.02 |   14.981 |    21.104

Write IO
thread |       bytes     |     I/Os     |     MiB/s   |  I/O per s |  AvgLat  | LatStdDev |  file
-----------------------------------------------------------------------------------------------------
     0 |        98250752 |        23987 |       3.12 |     799.46 |   17.826 |    20.308 | c:\DiskSpd\testfile.dat (10MiB)
     1 |        87617536 |        21391 |       2.78 |     712.94 |   20.059 |    21.165 | c:\DiskSpd\testfile.dat (10MiB)
     2 |        99315712 |        24247 |       3.16 |     808.12 |   17.986 |    20.387 | c:\DiskSpd\testfile.dat (10MiB)
     3 |        99102720 |        24195 |       3.15 |     806.39 |   17.900 |    20.380 | c:\DiskSpd\testfile.dat (10MiB)
-----------------------------------------------------------------------------------------------------
total:         384286720 |        93820 |      12.21 |    3126.90 |   18.396 |    20.565
```

# #9 Domino Disks

It also gives useful information on averages:

total:

| %-ile | Read (ms) | Write (ms) | Total (ms) |
|---|---|---|---|
| min | 0.034 | 1.037 | 0.034 |
| 25th | 0.317 | 3.478 | 0.416 |
| 50th | 0.501 | 5.027 | 3.524 |
| 75th | 41.544 | 43.991 | 42.250 |
| 90th | 44.460 | 46.767 | 45.713 |
| 95th | 50.911 | 53.609 | 52.020 |
| 99th | 59.778 | 61.788 | 60.764 |
| 3-nines | 61.085 | 65.299 | 63.932 |
| 4-nines | 98.420 | 98.682 | 98.682 |
| 5-nines | 99.793 | 100.229 | 100.060 |
| 6-nines | 99.807 | 100.229 | 100.229 |
| 7-nines | 99.807 | 100.229 | 100.229 |
| 8-nines | 99.807 | 100.229 | 100.229 |
| 9-nines | 99.807 | 100.229 | 100.229 |
| max | 99.807 | 100.229 | 100.229 |

Given a particularly demanding set of requirements on a server we can map out totals for all disks like this:

| Disk | Prescribed Latency | Actual latency (average) | Actual Latency 10 per cent of IO/s |
|---|---|---|---|
| C | Less than 3 ms | 16.345ms | 45.713ms |
| E | Less than 1ms | 6.954ms | 33.464ms |
| F | Less than 1ms | 10.023ms | 42.951ms |
| G | Less than 1ms | 9.997ms | 42.188ms |
| H | Less than 1ms | 6.96ms | 31.448ms |
| J | Less than 1ms | 9.991ms | 42.658ms |

# #10 Manage Full Text Indexes

A not uncommon server crash is can happen due corrupt full text index.  There's two things an admin can do to try avoid this:

-Recreate (delete and create) full text indexes relatively regularly. (i.e. even just yearly).

-Ensure appropriate antivirus exclusions are in place

There are few settings to manage and control FTIs, we'll go through them here.

The update frequency you set when creating the FTI:

Daily (needs to be managed by updall/dbmt)
Scheduled (needs to be managed by updall/dbmt)
Hourly (managed by update)
Immediate (managed by update)

# #10 Manage Full Text Indexes

It’s worth verifying is immediate actually always required?

To assess how update is currently managing, issue the following from the Domino Console. (which will tell you how many requests in the update queue have been deferred and are pending.)

*show stat Update.DeferredList*

*show stat Update.PendingList*

For immediate/hourly full text indexes it’s important to be aware that by default the update task will always process view updates first.  It is quite a CPU (and disk) intensive task so it can be helpful if you have the resources on the server to have a dedicated thread just for FTIs.

You can do this with the notes.ini setting *UPDATE_FULLTEXT_THREAD=1*  (however ensure the server has the appropriate resources to handle this)

# #10 Manage Full Text Indexes

You can change the directory/disk full text indexes are created on using the notes.ini *FTBasePath=* (however you should delete and create indexes on the new directory). Ideally these will be dedicated disks.isks that are optimised for this operat

To Disable FTI completely on a Domino server with the notes.ini setting *UPDATE_NO_FULLTEXT=1*

You can also manage how attachments are managed with FTI with *FT_Index_Attachments=<value>*

'1' attachments will be indexed.
'2' then attachments will not be indexed.
'3' then attachments will be indexed, and attachment filtering will be used if applicable.

# #11 Manage Views

Mentioned earlier but do review size of your views on an ongoing basis.

Within active/passive clusters make sure required views indexes on larger databases are built on passive node on an an ongoing basis.

For frequently updated all document views there are possibilities to optimise with dynamic indexing of high-usage views.  Using the notes.ini variable *NIF_VIEW_USAGE_ENABLED=1*

This will update the ten most updated views by a set of dedicated view threads.  Can increase performance 2-3 times on those views.
https://help.hcl-software.com/domino/12.0.2/admin/high_usage_views.html

Note the setting Debug_nif_update=1 can help you diagnose issues with particular views updating
https://support.hcl-software.com/csm?id=kb_article&sysparm_article=KB0037461

There are also the notes.ini option to move view rebuilds onto a separate (optimised drive), i.e. *view_rebuild_dir=*

# #11 Manage Views

Finally you have the option to move view indexes to another directory/path (ODS needs to be 51 or above and transaction logging must be enabled):

-Set the path with the notes.ini setting *NIFNSFEnable=*
-Compact on existing databases i.e. load compact -c -nifnsf on xxxxx.nsf
-To create for all newly created databases you need to also set the notes.ini setting *CREATE_NIFNSF_DATABASES=1* (and restart the server)

# #12 Keep your OS Admin in check

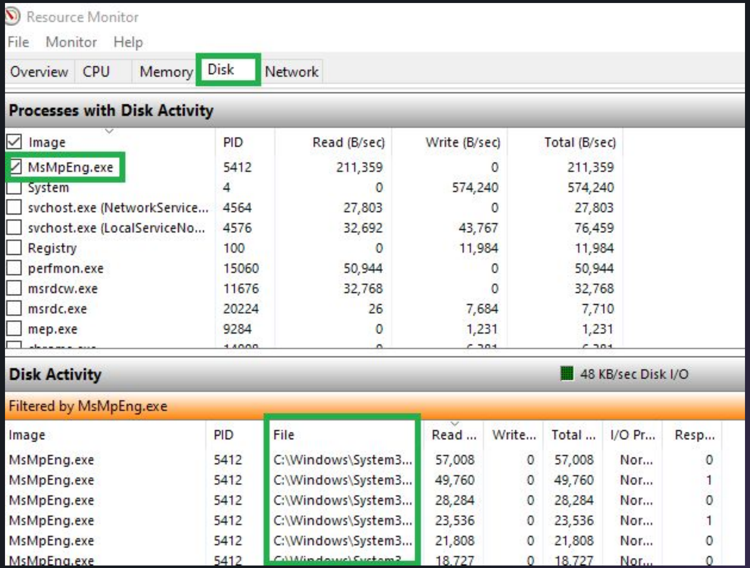
In controlled, regulated environments there's often delegation of duties.  So the host that Domino runs on is often not under your complete control. (controls notionally at least are a good thing).

If you want to ensure that your server is maintained properly you need to make sure "obvious basics" are covered by your OS Admin.

Appropriate Antivirus exceptions should be in place - https://support.hcl-software.com/csm?id=kb_article&sysparm_article=KB0093046

It can be relatively straightforward to detect if Windows Defender is scanning something it shouldn't.
Start-Run-resmon

Disk tab, filter by MSMpEng.exe (by ticking the box beside it in the first pane), then in the lower pane -check the File column.  Nothing related to Domino should be there.



# #12 Keep your OS Admin in check

Make sure the uptime of your servers (particularly in Windows) is never much over 30 days. I.e. make sure Regular OS reboots are in place. (should be in place for Windows updates anyway).

Where you have clustered servers, make sure this is taken into account for Windows Updates. I.e. do one node in a cluster one day, another another day.

Make sure backups complete before the start of the working day.

# #13 Backups

Just mentioned in last slide but can't overstate it enough - don't let you backups overun into the next working day,  Also ensure it doesn't interfere with nightly agents or maintenance.

Options already mentioned to help here including, archive old data, nifnsf to another directory.  There are other options like DAOS to consider.  DAOSTune can be your friend here.
https://help.hcl-software.com/domino/14.5.0/admin/admn_daostune_cmd_line_args.html

Other options include - get faster/better backup tech.  Domino is an Enterprise Software Suite.  The infrastructure it interacts with should reflect that.

De-duplication - do you really need to backup every database on every node in a cluster (or even if not clustered)? While also being mindful of DR scenarios here.

# #13 Backups

If you are running DBMT or another type of copy style compact – note the DBIID of a database changes.

Why is that important – if you are doing incremental style backups with archival transaction logs – you need to consider this.

Native Domino Backups are a consideration (particularly on small sites) in terms of an all in one, simple to deploy native backup solution

BackupVSS should be another serious consideration when considering a new backup solution particularly for integration with VEEAM.

It turns Domino into a native VSS writer in Windows.

# #13 Backups

All you need to do on a Windows 2019(or later) server running Domino 12.0.2(or later) is “load backupVSS” This registers Domino as a VSS writer on the OS. Now you just add that task to servertasks= so it runs at startup.

And then at a high level – get you snapshot Software (eg VEEAM) to look at your server hosts, and you can configure, native, application aware snapshots. (saving backup times, disk costs, and is generally much more efficient)

# #14 Old Fashioned Standardised Checks.

Should be a fixture in every medium to large site.  A checklist form for an admin to fill in at intervals.

Multitude of different checks/tasks on a schedule – Daily, Weekly, Monthly, Quarterly, Yearly

Domino is a good platform to create a database to manage all your checks.

The should be specific to every site (i.e. there isn't one catch all set of checks that works everywhere).  Knowledge of the environment, ongoing known problems and what issues are important to the the business (particular metrics to be collected for example) must be considered.  They also should evolve over time.

There should be a clear responsibility of who is completing the check on any given day. It doesn't need to be the most Senior Admin, but anyone completing the checks should know what to do if the encounter certain scenarios (even if that scenario is escalate to a Senior Admin).

The checks should be transparent.  I.e. there should be a record of the check, stamped with the time/date and who completely the check.  You should be able to view all records and there should be a culture of accountability.  It shouldn't be an exercise in box ticking.

# #14 Old Fashioned Standardised Checks.

Sample of items on a daily check:

-Servers Up/Down
-Mail Routing (dead mail etc)
-Server Monitoring tab for tasks – up/down
-DDM – review, record and address events
-Adminp errors/Events requiring Admin Approval
-Any events from events4.nsf generated to notify admins
-Resolve Replication Or Save conflicts in Domino Directory
-Any application driven checks specific to site (for example document count checks on geo dispersed apps that may have had replication problems, agents not running, or some other business critical workflow).

Sample of items on a weekly check:

-Check for expiring certificates
-Ensure Domino Servers have been restarted within required parameters (notify whoever if not)
-Check Notes/Domino Releases – note if any new have been released or planned.
-(if no Symmetric Clusters) – do a cluster analysis for missing replicas

# #14 Old Fashioned Standardised Checks.

Sample of items on monthly check:

-Generate required business reports (ACL changes to critical apps, Up/Downtime reports, activity reports)
-Review tickets for this month for linked incidents to known problems.  Trend to be addressed?
- Any infrastructure requirements for next month (more disk, new servers etc)
-Review who has Admin access to servers

Sample of items on more long term quarterly/yearly check:

-Application specific tasks - changing year on keyword docs, creating yearly archive etc.
-Define planned infrastructure projects
-Review sizing of environment.
-Refine checks
-Review ticketing trends over year
-Review and refine responses to any major incidents over the time period

# #15 Keep your Developers in check.

Slightly tongue in check title – we are all on the same team but there should be controls around applying development changes.

If resources allow and the environment is large enough to justify it – You should have a dedicated Development and UAT environment.

The best practice is that “significant” Design changes to important business applications should be applied through a template by an Admin. (ideally going through formal change control processes)

Being pragmatic it should be still possible for developers to perform reasonable day to day application maintenance.

# #16 Minimise who can do silly things

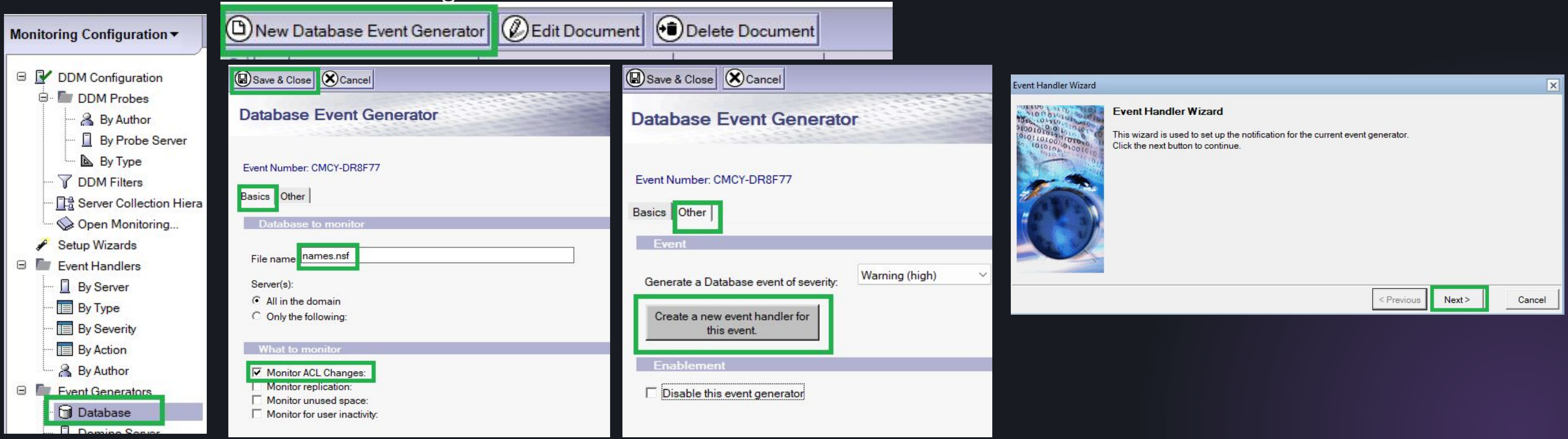
Restrict and control access to:

-names.nsf (author access to */org or via groups)- also check who has access to roles.

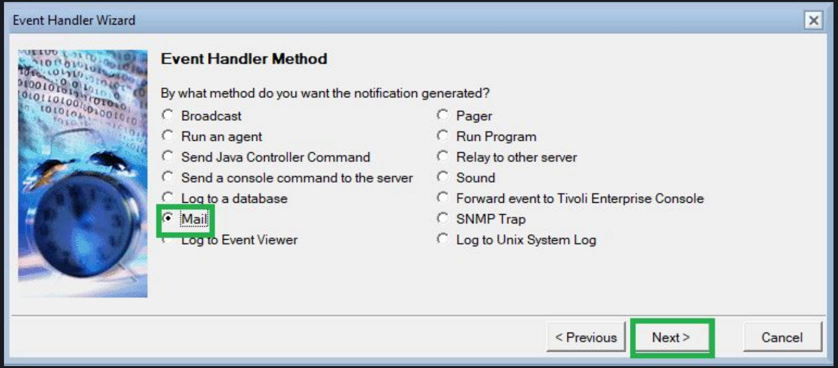
-Server Admins and signer privileges in Server documents, as well as the Admin ECL

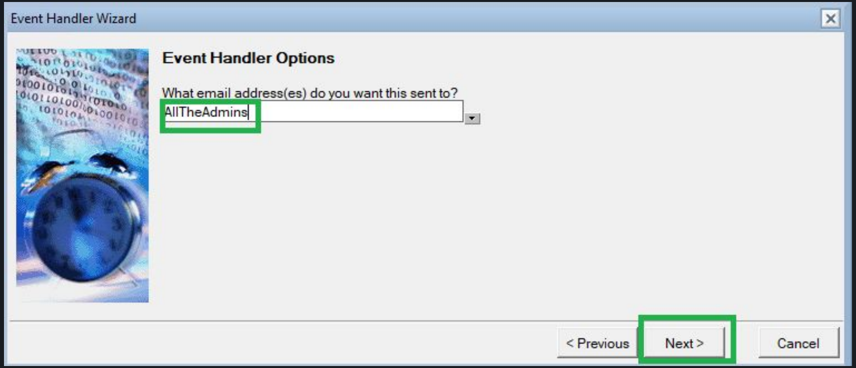
- system databases ACLs- “no access” should be default for those that don't involve authentication. (also set “do not allow URL open” by default)

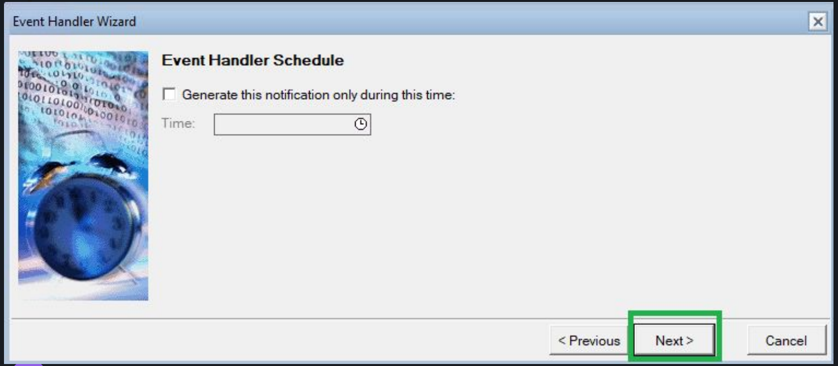
Set a events4.nsf alert if the ACL changes:

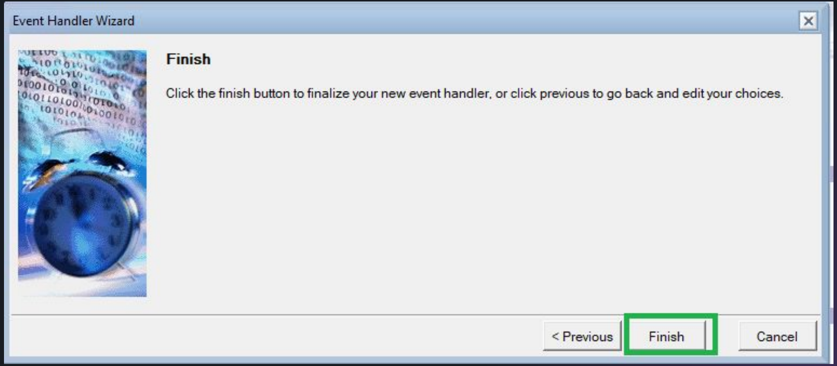


# #16 Minimise who can do silly things



# #16 Minimise who can do silly things

Designer and above access to production databases should not really be required for end users ever.

If you don't already -  pivot away from editing ACLs directly as a "day to day" practice in applications.

Standardise access via groups depending on level of access. e.g. one group for reader access, one group for editor, one group for editor with delete, one group for a particular role and editor etc

Ideally either the helpdesk or actual application owners can edit group members themselves without needing to touch an ACL. (or annoy you as an admin).

It also means you're minimising who has manager access to databases (i.e. the helpdesk or application owners shouldn't need it)

i.e. You can give ownership of access to ACLs without giving Manager access to ACLs.

- -Create Appropriate groups
- -Add Groups to ACL (eg SalesApp_Readers)
- -Add Application Owners/Helpdesk to Owners/Administrators of Group
- -Edit Helpdesk/Application Owners access in ACL. (potentially remove)
- -Educate Helpdesk/Application Owners on editing Groups

# #17 DAOS Maintenance

There's some maintenance options within DAOS every Domino Admin should be mindful of.

The first I would recommend as a default in most environments where there are a significant number of databases with DAOS turned on, would be to initiate a proactive program document to run a resync of the catalog once a week out of hours. This resynchronizes DAOS enabled databases with DAOS objects in the storage repository.

**Program: nserver**

Basics | Administration

| Basics | | Schedule | |
|---|---|---|---|
| Program name: | nserver | Enabled/disabled: | Enabled |
| Command line: | -c "tell daosmgr resync" | Run at times: | 05:00 each day |
| Server to run on: | DominoServer/Org | Repeat interval of: | 0 minutes |
| Comments: | Weekly proactive resync for DAOS | Days of week: | Sun |

This should also be a task that is initiated (again out of hours) if you restore a database.

# #17 DAOS Maintenance

- Earlier in the session Cluster Symmetry was mentioned.  If Cluster Symmetry isn't in place DAOS repair is a good option to look at. (assuming there's a replica of the database on another server in a cluster).

- Since Domino 14 there is a very handy command to run in the case where DAOS had done a resync and can't find a Domino object.   You can run a command for Domino to look for the object on other cluster nodes with:

*tell daosmgr repair objects*

Being pragmatic and proactive with maintenance – if you are already running a resync out of hours – it may make sense to run the above command via program document a couple of hours later on a schedule via a program document.

Full information is here - https://help.hcl-software.com/domino/14.0.0/admin/repairing_missing_daos_objects.html

# #18 Traveler Maintenance

Traveler is one of those services that generally just runs in the background generally without having to worry about it.

Since 14.5 – there are no more date based APNS (it's token based) – but you should still always keep up to date on every release.

One thing can happen [for non enterprise (HA) instances] is fragmentation of the database Traveler relies on to function. ie. the internal Derby relationational db (data\traveler\ntsdb )

Sometimes if there updates to traveler it will automatically defrag when you're updating.

You've multiple options to maintain it beyond that proactively:

Quit traveler and run:

```
load traveler -defrag
```

You may need to start http manually after the defrag is completed.

# #18 Traveler Maintenance

You can also run: `tell traveler dbmaint set interval 30` which runs a defrag on an interval of every 30 days.

Finally: `tell traveler dbmaint run` which forces a defrag the next time traveler runs.

Similar maintenance be completed on enterprise databases – (including PostgreSQL and MariaDB)
https://help.hcl-software.com/traveler/14.5.1/maintainingtheenterprisedatabase.html