HCL Software Academy
for Digital Solutions

HCL

Domino 12 Backup & Restore

Daniel Nashed
HCL Lifetime Ambassador

HCL SOFTWARE

Agenda

- Why Domino 12 native Backup
- Main Features and basic setup
- Linux, S3 and other options
- **Snapshot Backup** in Domino 12.0.2
- 3rd Party restore
- Questions and **Answers**

Domino 12 Backup
Introduction & Functionality

Domino Backup Requirements

- Domino **NSF** files need to be backup-ed on-line
  - Open file backup is **not** an option and leads to **inconsistent databases**!
  - You either need a Domino aware backup software
  - Or shutdown your Domino server for backup!

- VSS (Volume Shadow Copy) or file-system snapshots on Linux **alone** are not supported
  - A Domino Backup integration needs to support **Snapshot backup**

- Customers ask for archive style transaction logging to allow point in time restores

- New requirements like backup Docker containers and also new backup vendors
  - For example Veeam leveraging snapshot backup

Domino V12 Design Goal

- Full featured Domino Backup integration to bridge between Domino and backup solutions
- Allow integration with any type of backup vendor via custom scripts
- Allow Domino to integrate with customer's favorite backup solution

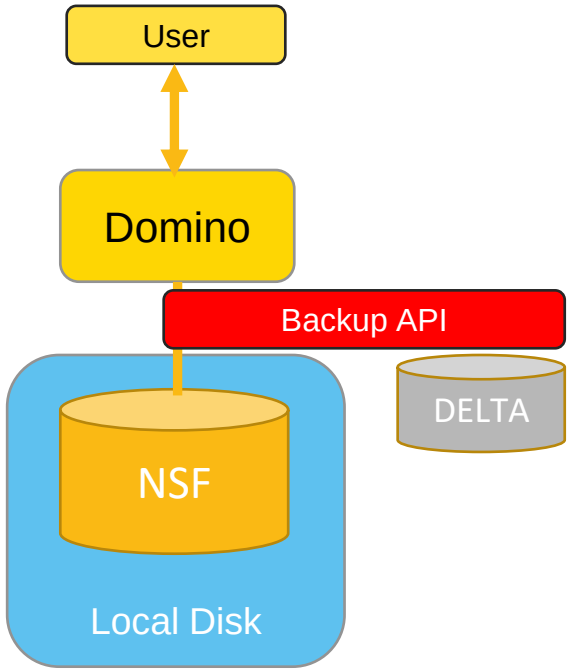
- Support **snapshot backup** applications
- Allows **broad** functionality on the Domino backup/restore side
- **Notes/Domino style UI** in combination with favorite backup solution

Main Functionality

- Flexible restore options
  - Bring database online, point in time recovery,
  - Disable replication, change replica ID, change title, disable all agents, etc.
- Command-Line/REST API, etc. integration for backup tools with file and snapshot backup
- Support for **Win64** and **Linux64** only
  - AIX and OS400 have strong IBM vendor support already
- Recover **documents** and **folders** into original database
- Integrated DAOS restore missing NLOs leveraging an existing file-backup
  - Focus is NSF & Translog backup. DAOS is a single file backup handled by any backup application

Technical Background

- **Problem**
  - Domino is constantly accessing files on disk; thus, file copy operations are causing inconsistent backups.
- **Solution**
  - **Domino Backup API** brings database into consistent state
  - Now NSF/NTFs can be backed up on file-level
  - Write access is still available to end users.
- Changes (aka deltas) are captured and **must** be applied to the backup to bring the database into consistent state!
  - Delta files are merged during backup in case of file backup operations
  - Or on restore in case of other operations
  - The Domino backup & restore takes care of tracking delta files



Backup and Restore Application Components

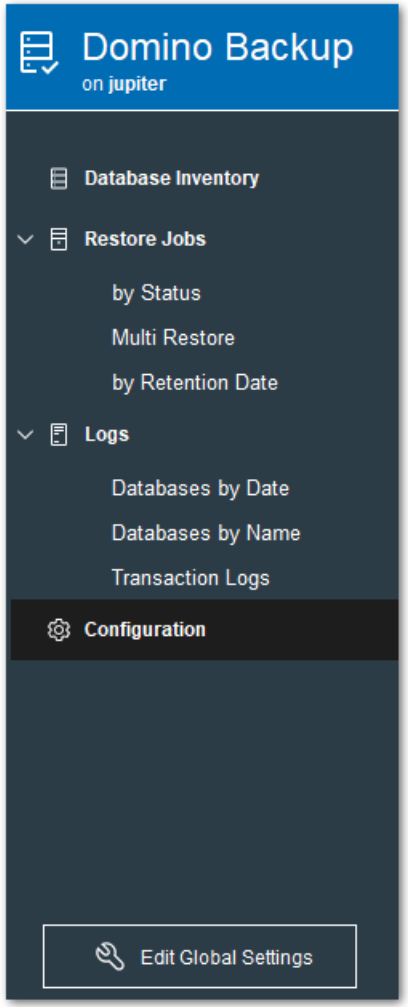
- **backup** – Backup servertask invoked via program document

- **restore** – Restore servertask either
  - a.) invoked on command line or
  - b.) running permanently monitoring the restore job database via -g option

- **dominobackup.nsf**
  - Configuration
  - Database Inventory and Backup Logs
  - Restore interface and restore jobs

dominobackup.nsf

- Configuration
  - Per server or global configuration per platform
  - Global settings
- Per Database Backup log repository
  - Showing all backups per database and base for restore operations
- Per Backup log for NSF files and Translog
- Restore requests
  - Derived from per database backup, when you select a restore



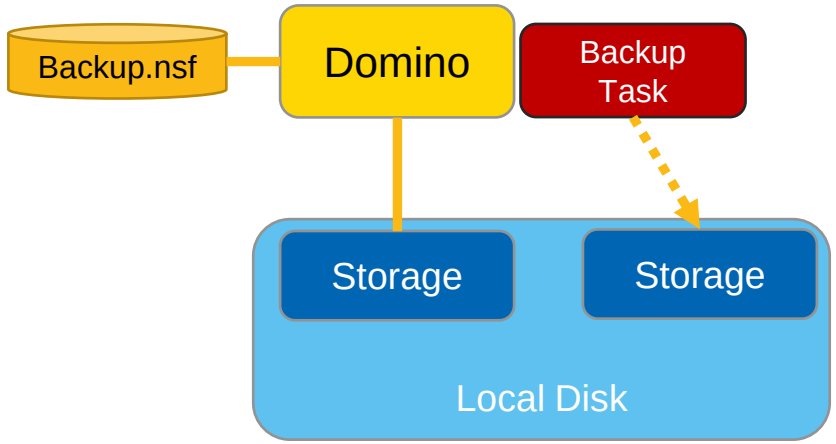
Backup Concept : Full Backup

- Full backup only
  - Allows to recover data at the point in time when the backup was taken
- With circular Transaction logging is enabled in Domino
  - Limited point in time recovery if transaction logs are still on disk
- Typical setup:
  - Daily Full backup
  - Backup job running outside of business hours

Backup Scenario: Disk / File Share

- **Configured out of the box**
  - Just needs a “disk” with sufficient space to backup databases
  - Very simple to setup
- Databases changes during backup are automatically merged into the database backup!
  - Consistent backup without the need to use the restore servertask
  - Databases can be just copied back – Don’t overwrite existing databases on OS level when the server is up!!
- Any type of **file-share**
  - **Recommended**: Storage supporting compression and de-duplication
    - **Cohesity** → Backup share into backup repository (called “View”)
    - e.g **TrueNAS** → https://www.truenas.com/ - leverages ZFS storage like many others on the market
    - Other deduplicating storage like **NetApp** etc..

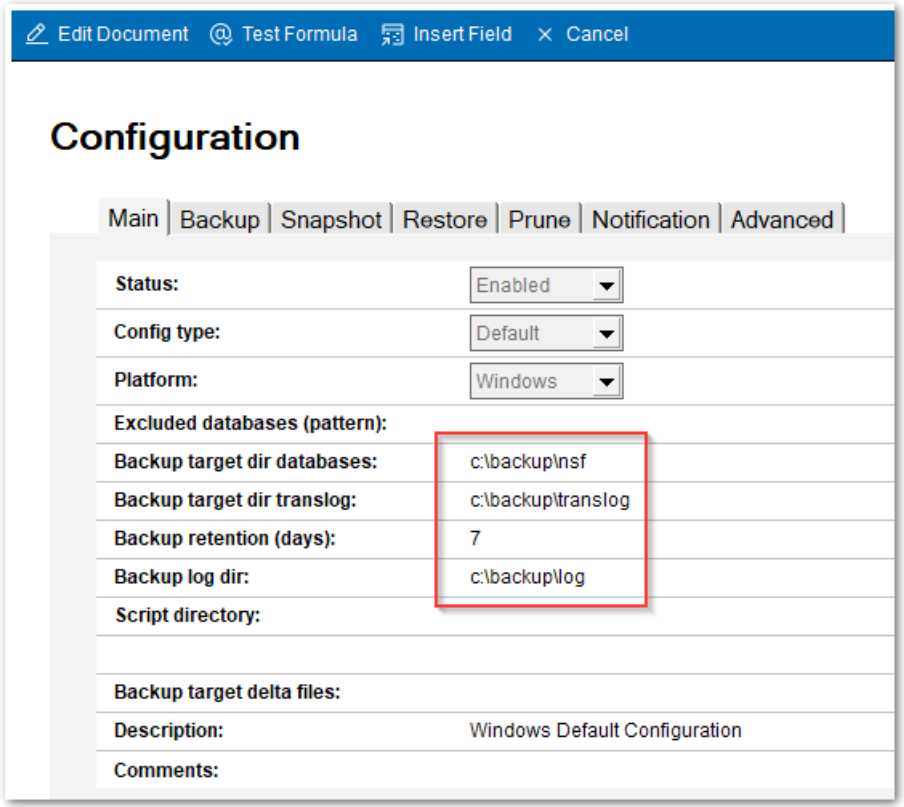
Quick Start Backup

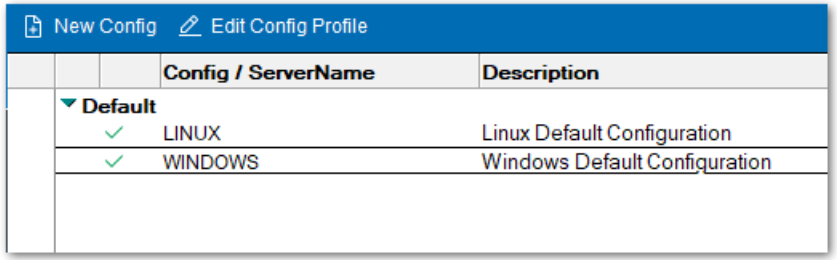


- **1. load backup** → creates new dominobackup.nsf

- Review configuration
  - One pre-defined file copy configuration per platform
  - Already enabled
  - Check if the target location works in your environment
  - Server needs write permissions to create the directory!

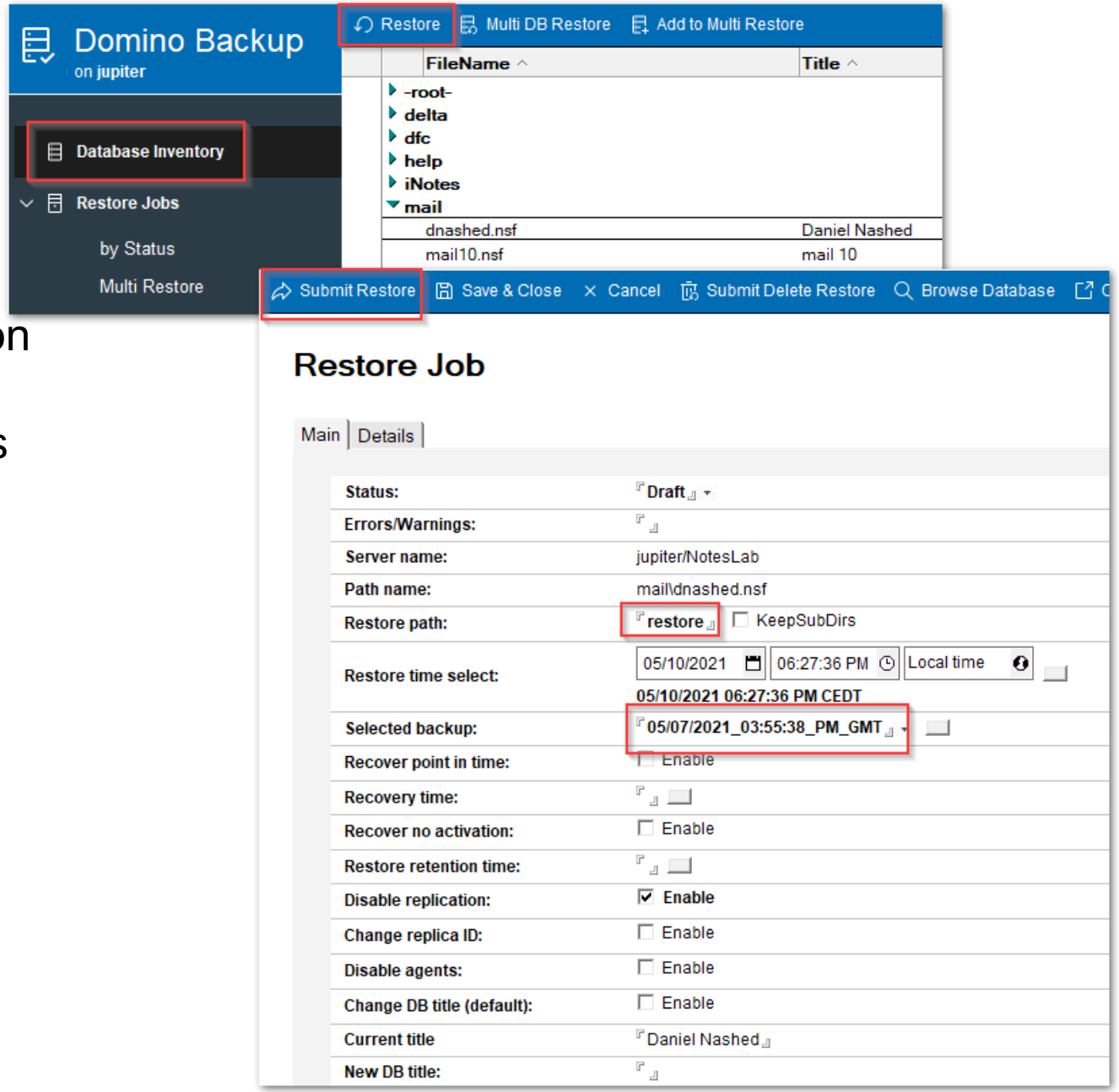
- **2. load backup** → to start your first backup





Restore Operations

- 1. Find the database to restore
- 2. Create a “restore job” using the action button
- 3. Specify the restore time and restore options
  - Most important options are set by default
- 4. Submit the restore using action button
- **5. load restore**
  - Tip Load restore -g
  - Allows to keep the restore application loaded
  - If you have remote console permissions:
    → Action button to start restore

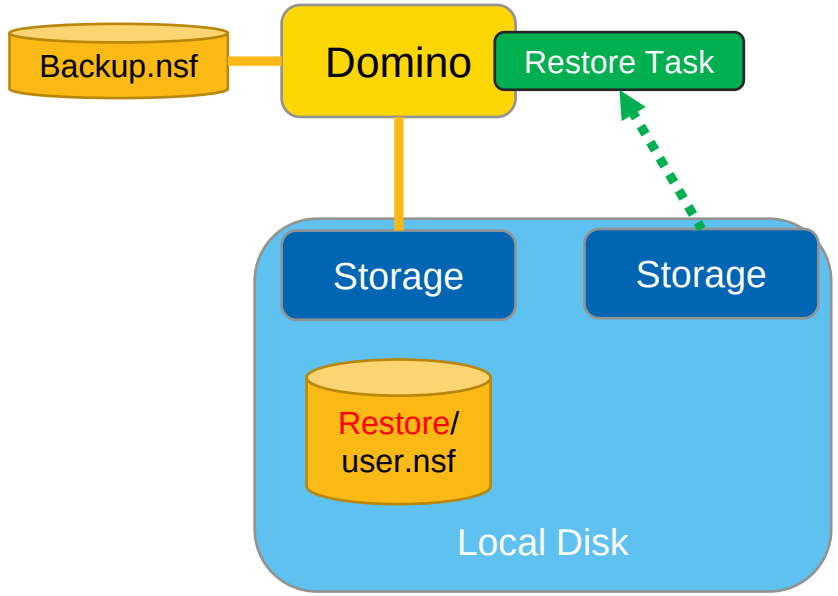


File Types involved

- ***.nsf, *.ntf, *.box**
  - Original databases to be backed up
- ***.txn**
  - Transaction log files to be backed up
- ***.DELTA**
  - Deleta files created during backup – if database changes during backup (e.g. names.nsf.DELTA)
- ***.DAD**
  - Restored database before “activation” (e.g names.nsf.DAD)
  - Restored with a temporary name to avoid server will find the database before it it recovered

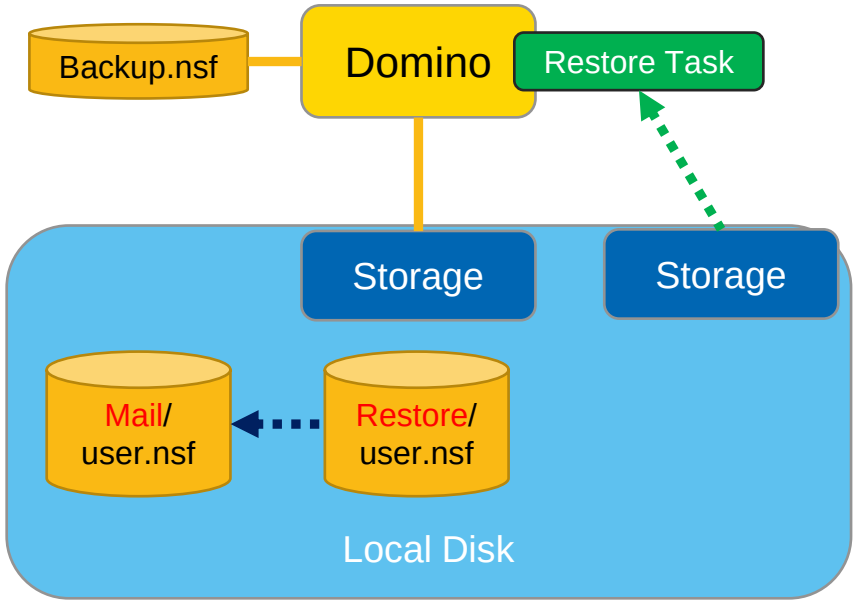
Restore operation in detail

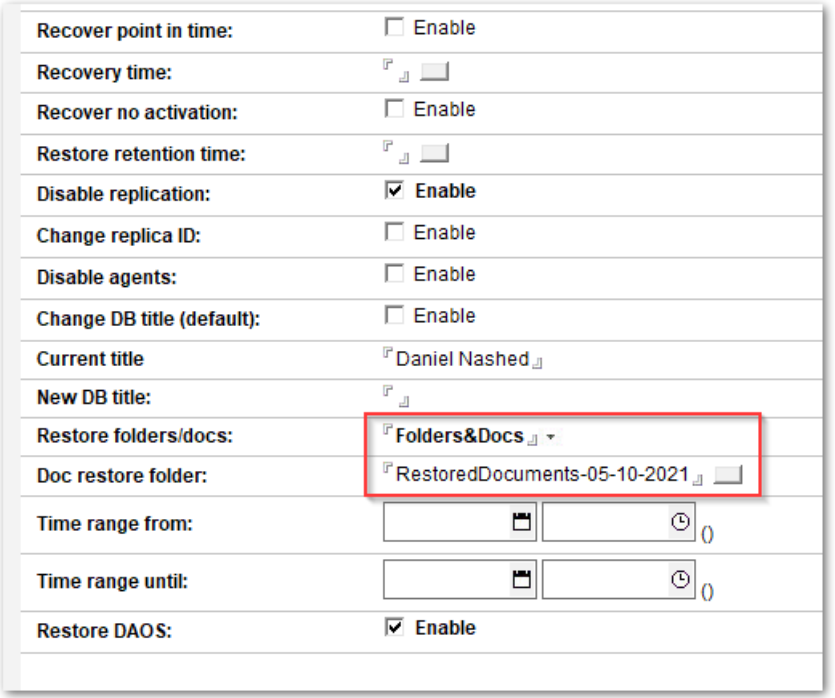
- Servertask copies back database to restore location
- File name will have a **.DAD** extension to ensure the server does not access it meanwhile
- Disables replication and sets other settings
- Applies **.DELTA** file if available and recovers the database
- Renames the database to remove the **.DAD** extension
- Recovery documents and folders into original database if requested



Restore documents & folders to original database

- **Restore documents and folders**
  - **Restore** deleted documents & folders into original database!
  - Finds deletion stubs and updates it to replace the deletion stub
  - Folder notes are updated and replicated back to the original database overwriting the deletion stub





Backup Integrations

Integration Points

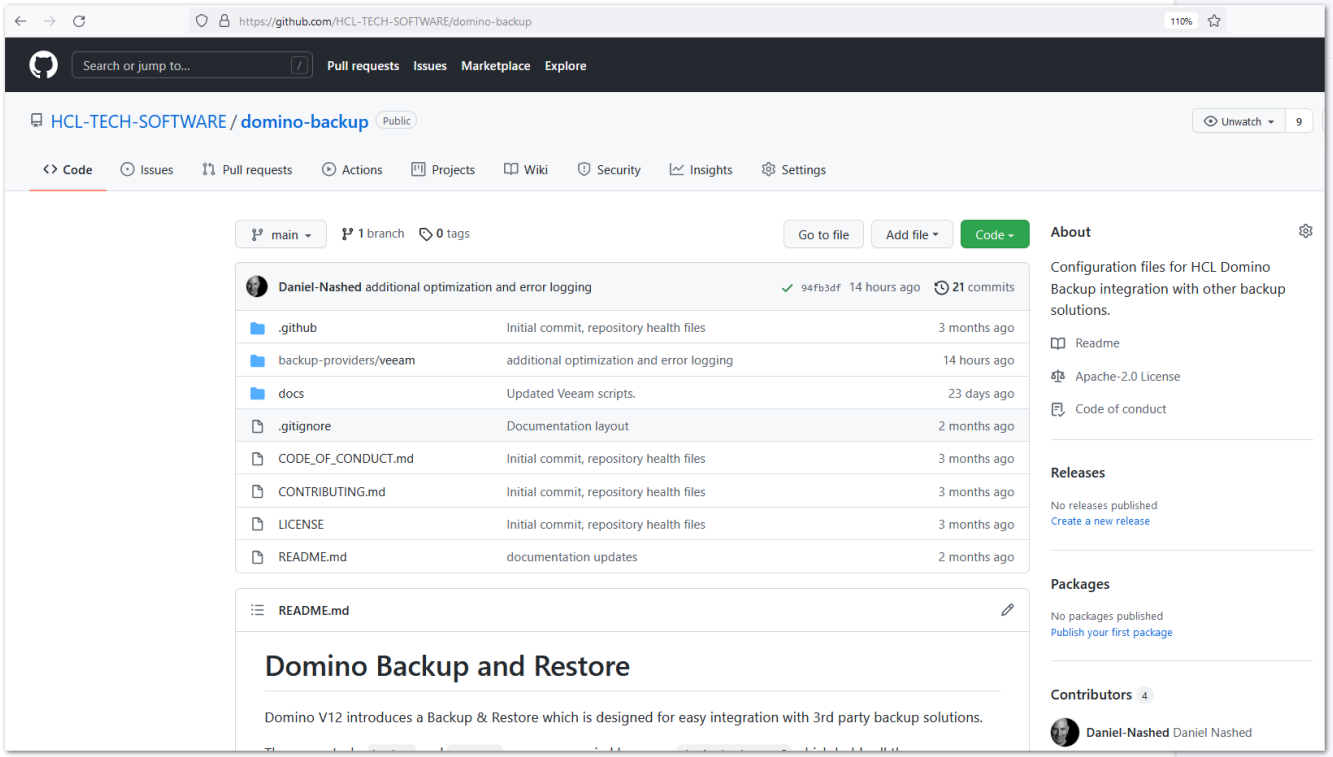
- **File backup commands** – build-in, leveraging OS commands
  - Most easy to setup backup/restore
  - Customizable via @formulas
  - Can be also leveraged for commercial software like Cohesity with their backup repository
  - Used for backup, restore and backup prune operations (delete)
- **Command-Line interface**
  - Very flexible, standardized interface for Windows (batch) and Linux (shell script)
  - Customizable via @formulas
  - Return strings can be passed to the backup software for backup reference and status
- **Agent interface**
  - Can be for example used for REST requests leveraging HTTP Request class

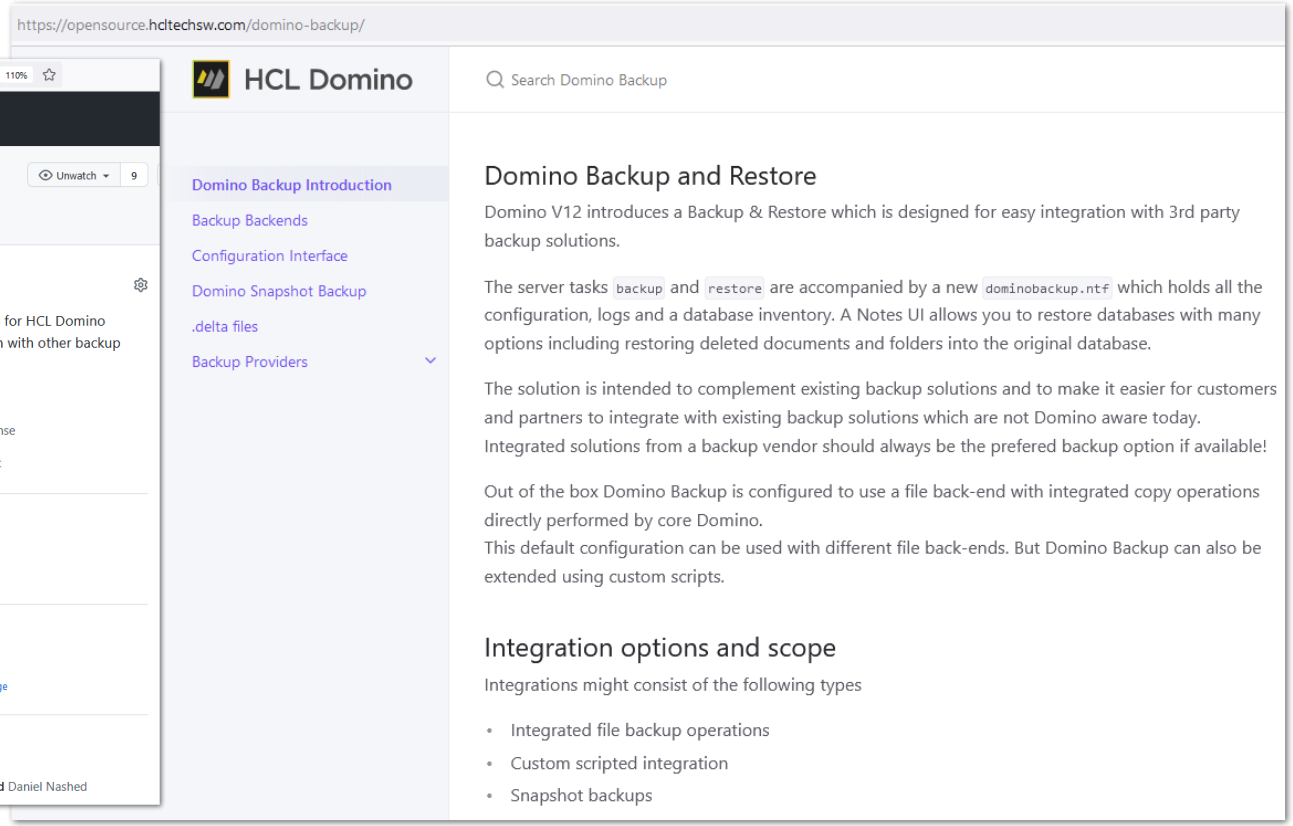
Backup Scenario: Integration on Script level

- **Command Line backup with a backup application**
  - For example: Linux Borg Backup → https://borgbackup.readthedocs.io/en/stable/
  - Or command line integration with a client like AWS S3 CLI or any other custom script
- **Flow**
  - Domino backup servertask takes one database after another into backup mode
  - OS level command is called to backup the database
  - Bring database into normal operations mode
  - Check if changes occurred during backup and create a delta file with changes
- **Challenges**
  - Restore operation is required to bring a restored database online
  - Deltas need to be applied back on restore

HCL GitHub Project for Backup Integrations

- https://github.com/HCL-TECH-SOFTWARE/domino-backup
- https://opensource.hcltechsw.com/domino-backup/





HCL Software Academy for Digital Solutions

HCL SOFTWARE

“Delta Files” created during backup

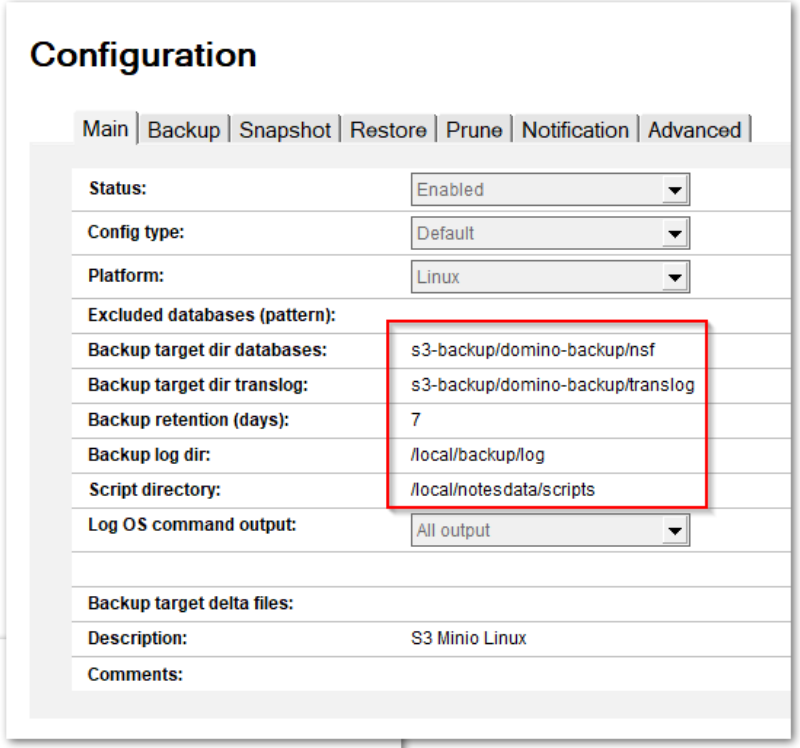
- The Domino Backup API brings databases into a consistent state for backup
- Databases can be backed up on file-level once backup is started
- **Important:** Changes in database are recorded and **need to be applied** to the backup database to bring the database into **consistent** state!

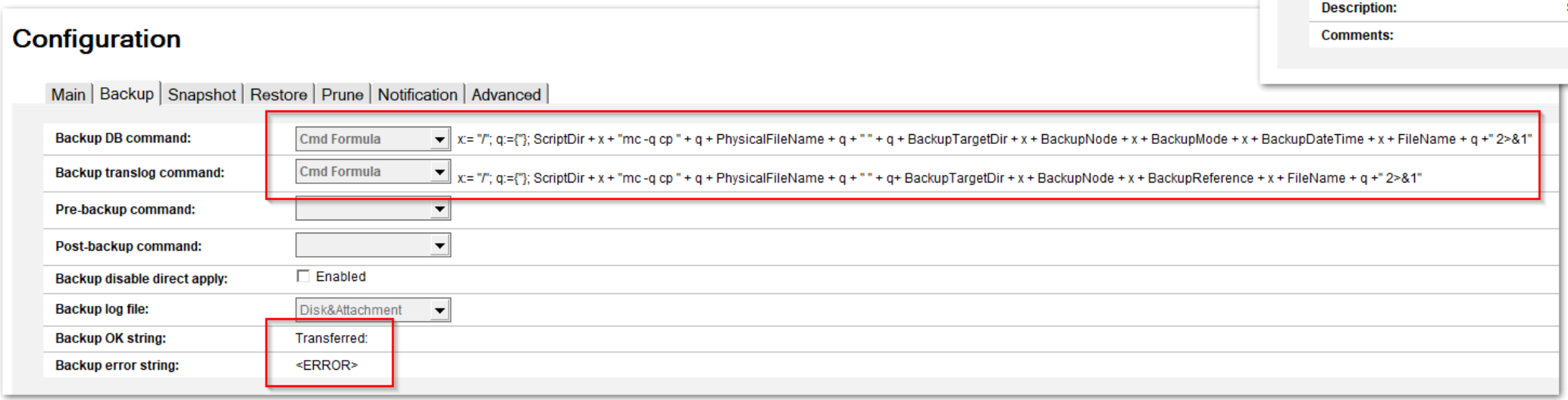
- Delta files can be
  - a.) Merged during backup in case of file backup operations
  - b.) On restore in case of other operations

- **Delta Files have to be merged in any case!**
  - But don't worry - The Domino backup & restore takes care about tracking delta files

Implementation reference: S3 Storage Minio

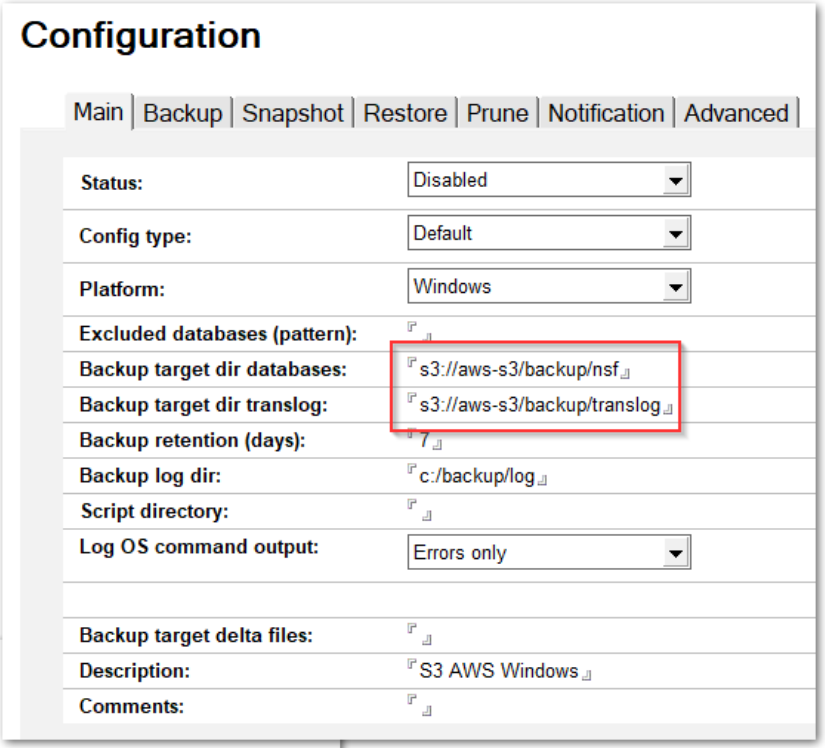
- Command line integration
- Simple integration based on **formulas** not a “Cmd file”
  - Good example to show how @Formula integration works
  - Results are captured by the backup/restore task
    - OK and Error strings can be used to check operation status
  - Needs helper binary and config from S3 vendor (e.g. **mc** from Minio)

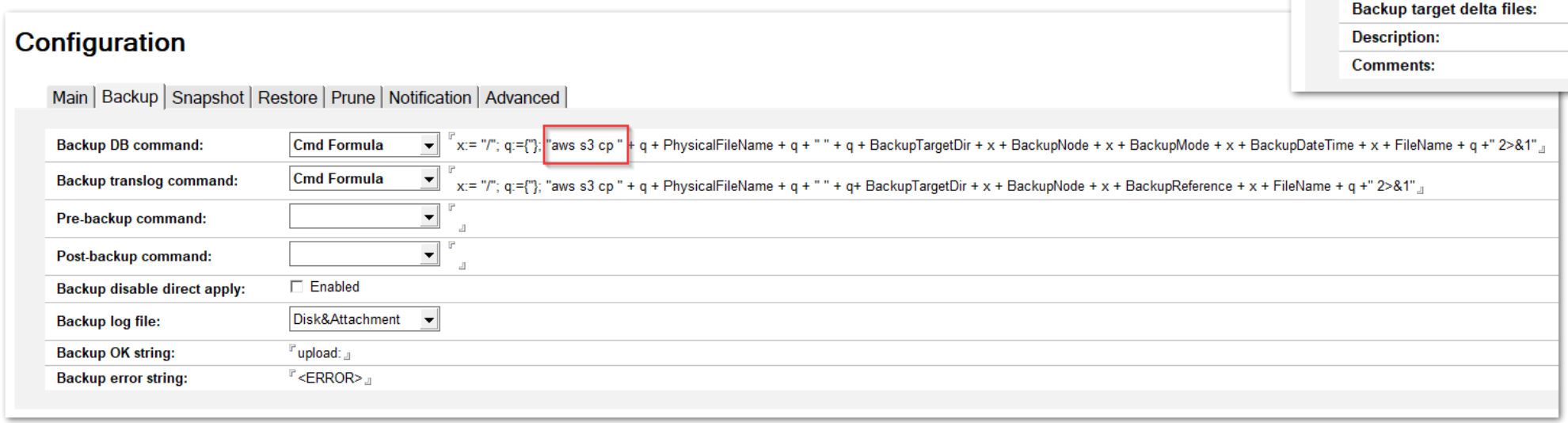




Implementation reference: S3 Storage AWS CLI S3

- Same type of integration
- Just a different command line used
  - In this case AWS CLI with S3 command
- Needs target specific configuration
  - AWS credentials and target information for the bucket used





HCL Software Academy for Digital Solutions

HCL SOFTWARE

HCL GitHub Project for Backup Integration Solutions

- Open Source repository for 3rd party backup integrations based on the **framework** Domino provides
- Integration point/operations documentation
- Additional information
  - Technologies like snapshot
  - Best practices around backup and storage optimization
  - File system back-end operations
- Main entry point for supporting backup integrations
  - It is not intended that HCL support should help to integrate applications
- Collaboration in the community via GitHub repository
  - **Backup vendors, customers, partners and HCL**

Snapshot Backup

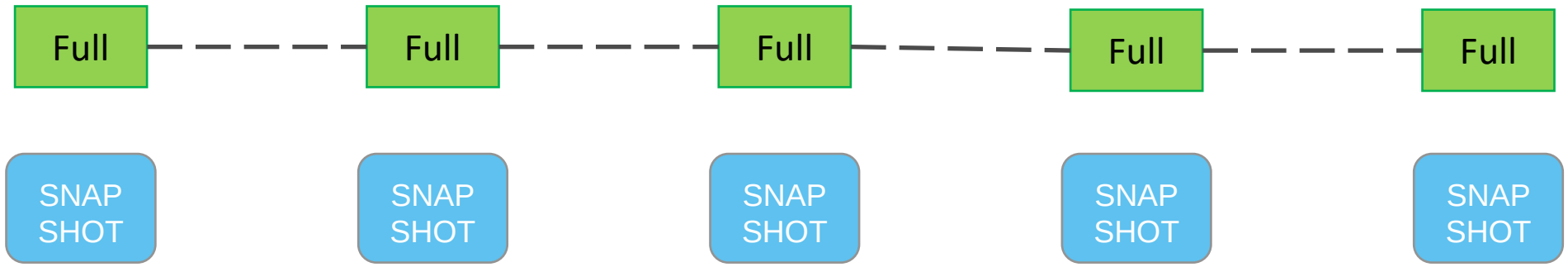
Background & New Features

Backup Scenario: Snapshot

- Works similar like previous scenario but brings all databases into backup mode at once
- **Flow**
  - Bring all databases into backup mode
  - Call an OS level command to take snapshot
  - Bring all databases into normal operations mode
  - Create a **delta** file with changes for each database with changes
  - Operations on Domino side finished, snapshot and delta files are static and any file backup can be used
- **Backup using snapshot**
  - Many different options depending on the solution used
  - Most simple approach on Windows – without a backup application supporting snapshots itself
    - Call a VSS snapshot command and to get a temporary snapshot
    - Use any file backup solution to backup the snapshot along with delta files

Backup Concept : Snapshot

- **Flow:**
  1. Backup application brings all databases into backup mode
  2. Snapshot is initiated – should not take more than a couple of sections
  3. Delta files from backup are stored separate from the snapshot
  4. Snapshot and delta files are backed up
- On restore usually the snapshot is mounted to “copy” databases back
- Delta files are usually saved to different backend or separate snapshot of another disk



VM Level Snapshots

HCL Domino

- Many backup vendors support VM level/storage level snapshots
- Main technology used
  - VMware VSphere snapshots leveraging their storage API
  - Hyper-V snapshots
  - Storage snapshots e.g. NetApp and others
- Already supported: Veeam backup leveraging VMware VSphere & Hyper-V backup
  - Flow:
    - 1. Veeam initiates backup to bring Domino into snapshot mode “**Freeze**” event
    - 2. Veeam generates snapshot and unfreezes databases with “**Post Thaw**” event
    - Implemented leveraging **integration scripts** for Windows and Linux to control “Domino Backup”

Linux Snapshot Support

HCL Domino

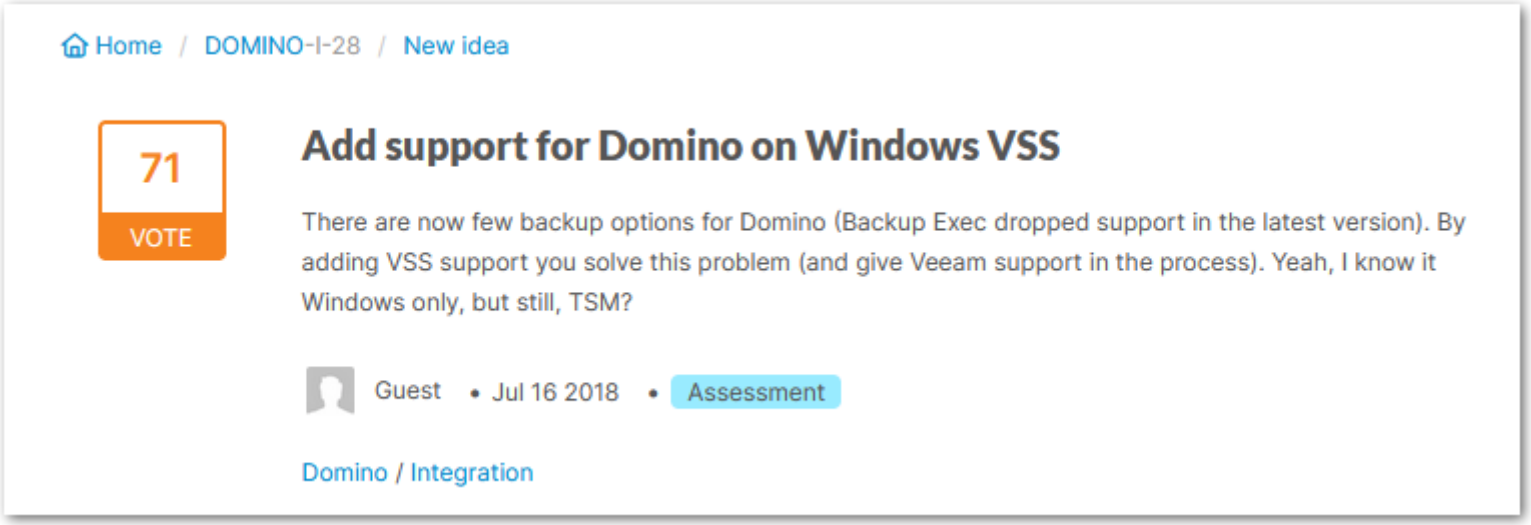
- No build-in snapshot functionality into Linux core
  - But available for certain file systems
- **OpenZFS**
  - Widely available for most distributions and supported by Domino
  - OpenSource project forked from Sun long time ago and very mature enterprise grade file-system
- **Btrfs**
  - Mainly used by SUSE and supported by Domino
- Both file-systems support snapshots created and managed by OS level commands
- Domino Backup invokes snapshot commands and manages snapshots
  - Create, Mount, Delete

What’s new in Domino 12.0.2 Backup

Native Windows VSS Writer Implementation

What’s next?

- The next logical step would be to make Domino a full native VSS Writer



VSS Writer Support

HCL Domino

- Volume Shadow Copy Service (VSS) supports application integrated snapshot operations
- “VSS Writer” allows to make an application fully “**snapshot aware**” without any direct backup application integration or scripting
- Requires Domino to become a “**VSS Writer**” supported application
- Flow
  - Domino registers as a “**VSS Writer**” using a Microsoft VSS API
  - Backup application starts VSS Snapshot
  - Windows sends event to all VSS Writers registered to “**Freeze**” their application
  - Windows takes VSS snapshot
  - Windows sends VSS “**Post Thaw / Post Snapshot**” event to application
  - Domino processed delta data accumulated during snapshot operations (60 < sec)

Domino VSS Writer Requirements

HCL Domino

- Domino needs to register as a native **VSS Writer** on startup
  - Leveraging the Microsoft VSS API
- Domino needs to permanently listen to **VSS call-backs** and handle "**Freeze**" and "**Post Thaw/Post Snapshot**" events to initiate Domino Backup Snapshot operations
- Domino needs to be permanently registered as a VSS writer and respond to VSS events
- Works with all VSS backup aware backup applications without any additional scripting for backup!
- VSS writer integration is designed as a **backup only integration**
  - Restore still requires an integration to "**mount**" snapshots to restore NSF files
  - This is in alignment with backup vendor implementations → VSS is just used as a helper technology

Domino VSS Writer

HCL Domino

- VSS Admin Windows tool shows all registered VSS Writers
- Once registered, Domino is listed as a backup writer

```
vssadmin list writers

vssadmin 1.1 - Volume Shadow Copy Service administrative command-line tool
(C) Copyright 2001-2013 Microsoft Corp.

Writer name: 'Domino Backup Writer'
   Writer Id: {b95d0c5e-57d4-412b-b571-18a81a16abba}
   Writer Instance Id: {287e5f15-b760-4024-9719-4b995206faf5}
   State: [1] Stable
   Last error: No error

Writer name: 'Registry Writer'
   Writer Id: {afbab4a2-367d-4d15-a586-71dbb18f8485}
   Writer Instance Id: {627f7844-3a6c-4202-a4d6-1886edbf5c06}
   State: [1] Stable
   Last error: No error
```

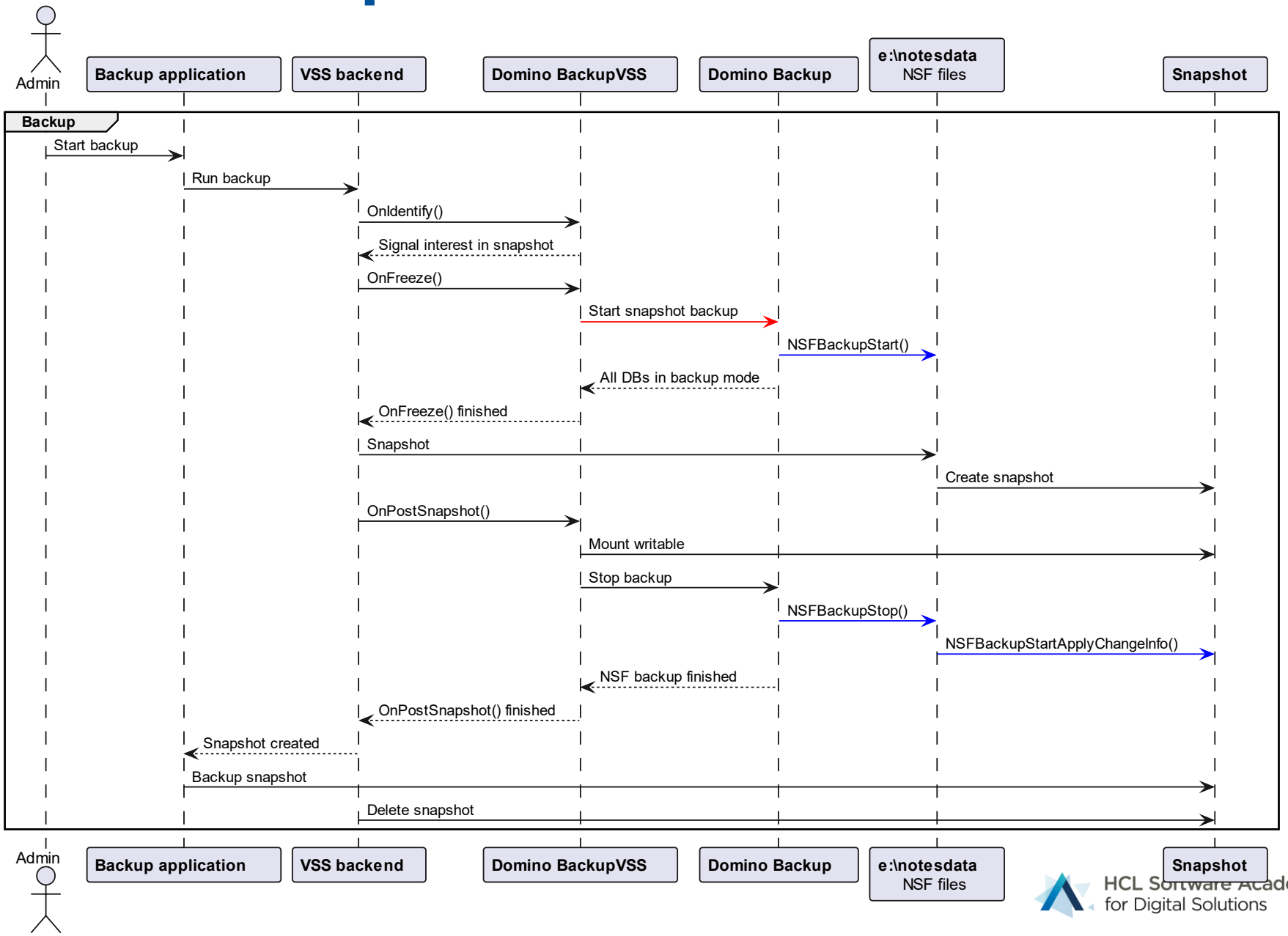
VSS Writer “AutoRecover” Support

HCL Domino

- The biggest challenge for snapshot backup
  - **Snapshots cannot be modified**
  - **Delta changes** need to be stored separately and need to be applied to the database on restore to make the NSF file consistent

- **Solution**
  - VSS Writers **VSS_VOLSNAP_ATTR_AUTORECOVER** Option
  - Allows a **VSS Writer** to update the snapshot in the **OnPostSnapshot** event to
    - Merge **delta** information occurred during backup
    - Make the database consistent for recovery without Domino restore operations
  - Tricky but small modification in the snapshot flow in **backup** servertask to apply changes directly into the writable snapshot in **OnPostSnapshot** event

Domino Backup VSS Writer Flow



Implementation

HCL Domino

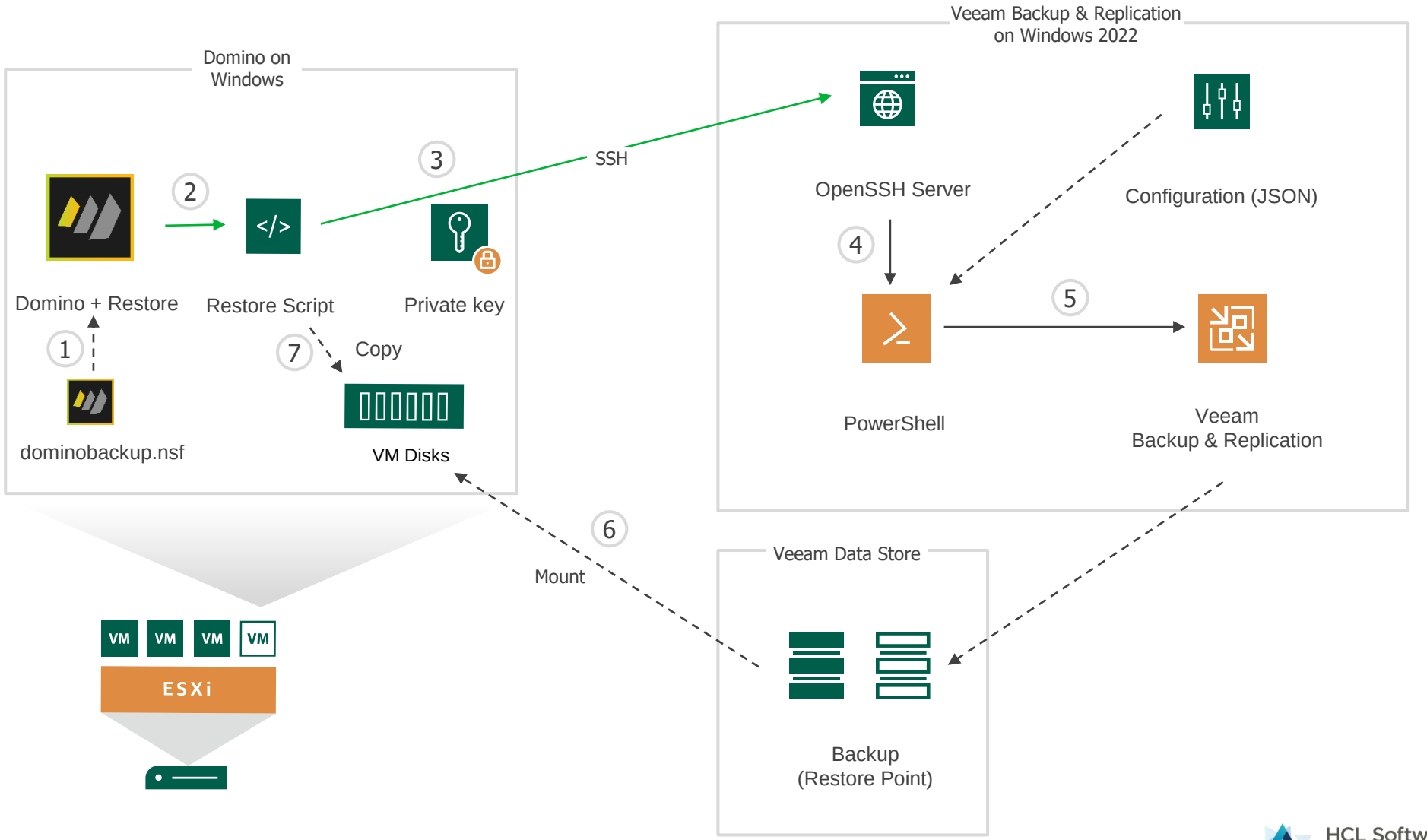
- Separate “**backupvss**” servertask registering as a **VSS backup** writer
- Invokes Domino Backup servertask (**backup -vss**) to leverage “Domino Backup Snapshot Mode”
- In **Freeze** event waits for backup task to bring all databases into backup mode
- Fully functional implementation of VSS API
  - Integrate **VSS Writer functionality** into Core Domino
  - Separate task is needed to control **backup** servertask
  - “**backupvss**” task is required to be permanently loaded to allow VSS to communicate with Domino

VSS Writer Implementation Limitations

HCL Domino

- NSF Data is required to be on a single volume for snapshot
  - No support for external directory or NSF links pointing to a different volume
  - No support for Windows junctions and comparable mount options
  - Support for directory and NSF link on the same physical volume
- VSS Snapshot backup application is required to support "**AutoRecovery**" mode for full functionality
  - Fallback to write delta files is still possible – In the same way it is supported in 12.0.1 today
- Restore integration still requires separate integration similar to Veeam integration available today
  - Restore integration scripts are posted in **GitHub repository**
  - No support for VSS restore operations
  - Vendors backup to their own repository and have no direct VSS restore integration
- Only one Domino partition per Windows machine can be backed up via VSS

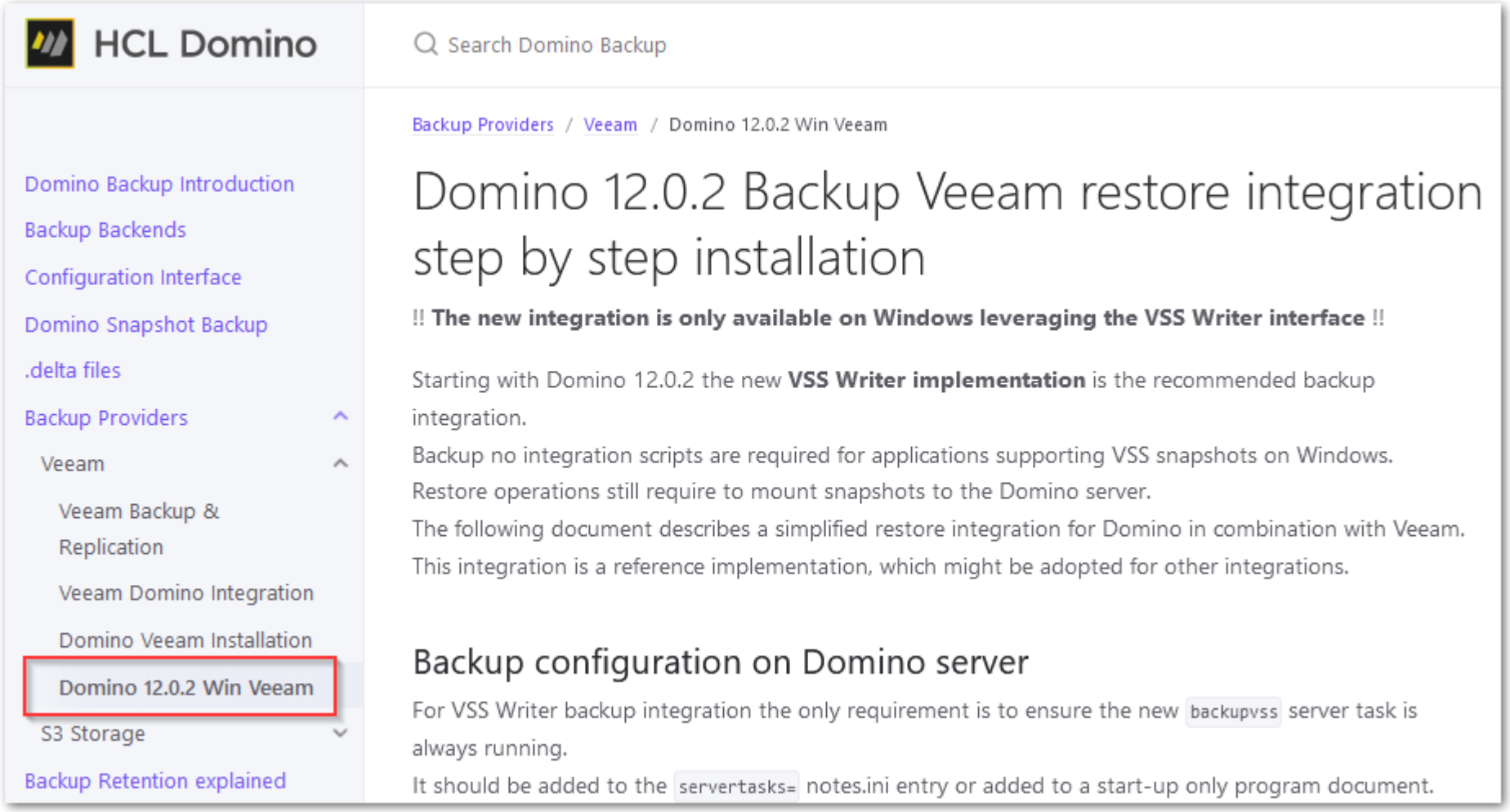
Veeam Backup & Replication Domino Restore



Updated Veeam Integration

HCL Domino

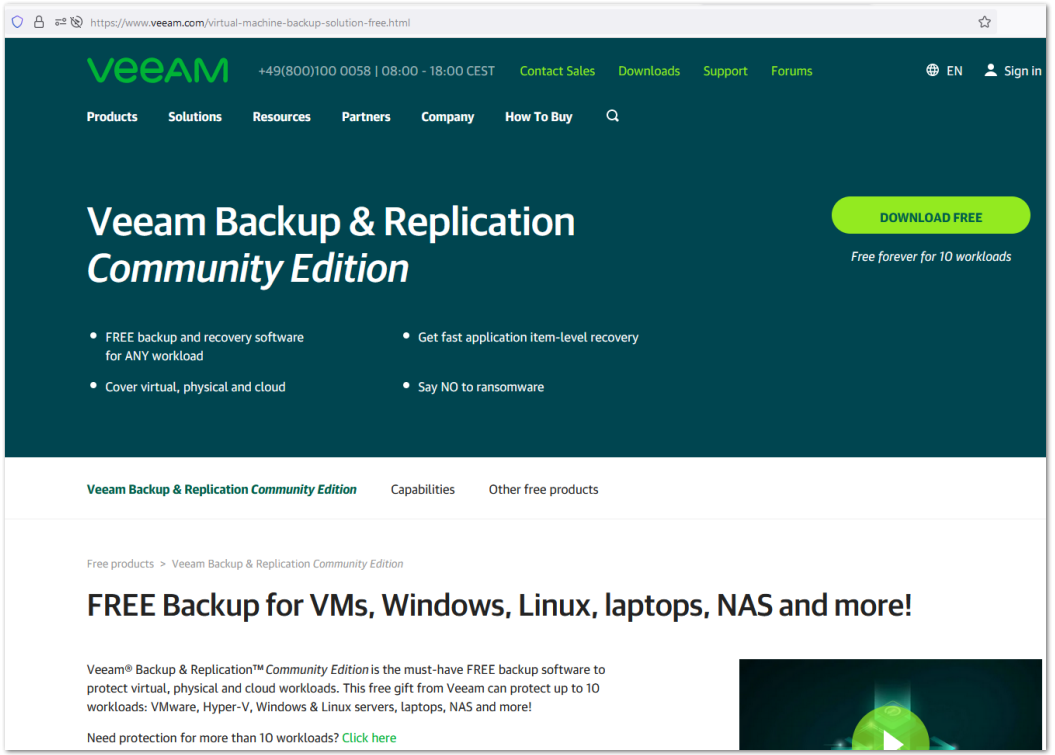
- Updated (and simplified) documentation for Veeam integration
  - https://opensource.hcltechsw.com/domino-backup/backup-providers/veeam/install_vss_writer/



HCL SOFTWARE

Build your own Lab Environment

- Free Backup for up to 10 Workloads!
  - Provides even free agents for workstations / notebooks
  - Can be used in production!

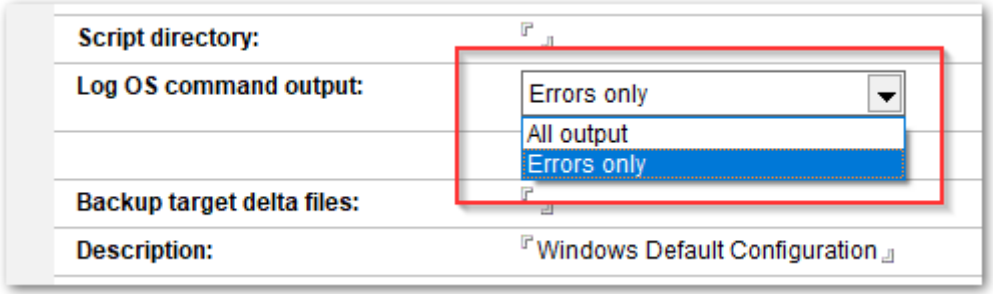


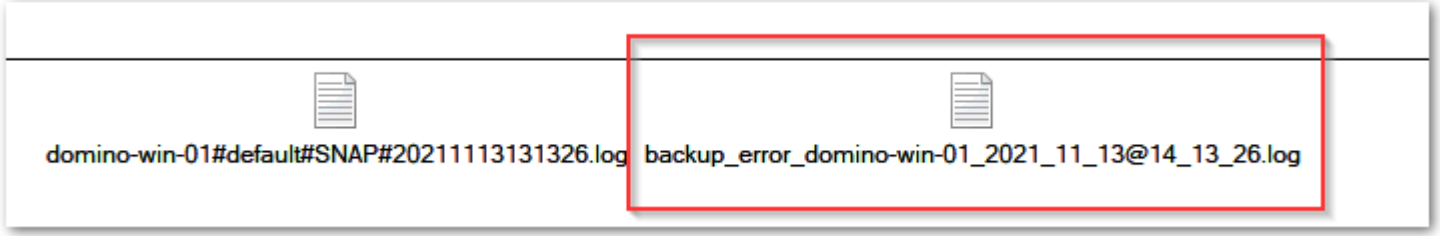
- Full featured functionality for virtual environments
  - VMware vSphere
  - Microsoft Hyper-V
  - https://www.veeam.com/backup-replication-system-requirements.html
- Community edition: https://www.veeam.com/virtual-machine-backup-solution-free.html

Domino 12.0.1
New Features & Improvements

Domino V12.0.1 End to End Logging

- In Domino 12.0 logs for command-line/script integrations are only parsed for errors
  - For debugging you had to write the logs into files and manage them on your own
- Domino 12.0.1 backup automatically collects all **standard output** from invoked scripts/ commands
  - When backup/restore scripts/commands fail, the output is stored in a temporary file
  - Once the backup or restore operations completes, the full logfile is appended to the backup/restore note
  - On by default and cannot be disabled, Option: log all output for debugging purposes (Main configuration tab)
  - Redirect error output for scripts by adding **2>&1** direct to the command configuration
  - Important: Redirect STDIN via **</dev/null** Linux and **</dev/NUL** on Windows





Domino V12.0.1 Multi Restore Improvements

- In Domino V12.0 the multi restore operation was limited
  - All databases had to come from the same backup
  - There have been some UI issues not showing the right error messages
- **Domino 12.0.1 now fully supports multi database restore**
  - Always the latest backups for a database matching the restore time, are used
  - Databases don’t need to be in the same backup (e.g. incremental backups)
- How does it work?

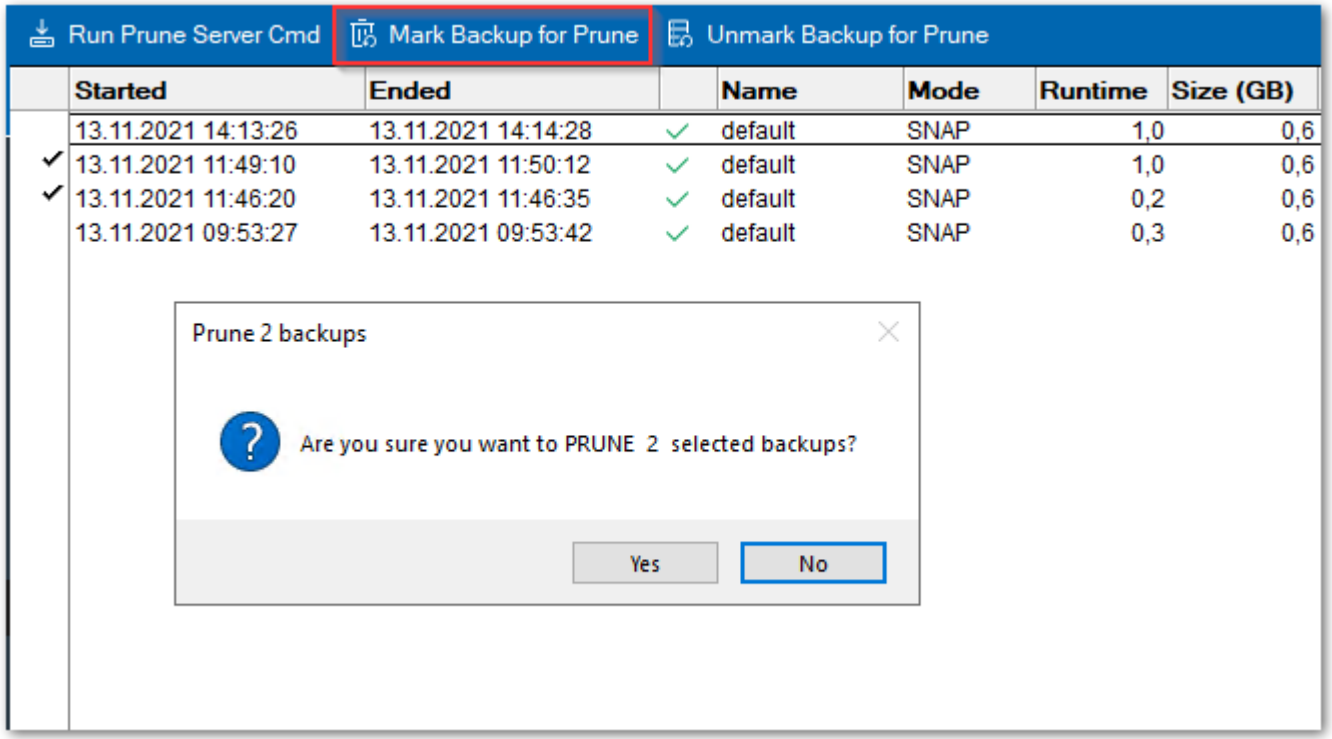
  1. Admin selects the restore time
  2. Submit logic checks each selected database for matching backup and writes the time into each doc
  3. Restore operation will restore each database from the specified backup date

Domino V12.0.1 Multi Restore Optimization

- **Challenge**
  - Restoring multiple database from a snapshot would potentially involve many mount/unmount operations
  - For many backup vendors mounting a snapshot is a quite time consuming operation
- **Solution**
  - New pre-restore and post-restore scripts to allow restore operation optimization
- Example logic:
  - **Restore DB operation script** → Checks if the right snapshot is mounted
  - → If yes, just copy the database from snapshot
  - → If not mount the snapshot and copy the file
  - Next file might hit the already mounted snapshot – or mounts another snapshot
  - **Post Restore Script** → Unmounts all mounted snapshots

Domino V12.0.1 Selective Backup Prune

- In Domino 12.0 backups are only pruned by retention time
- Domino 12.0.1 introduces a selected prune operation directly from the Log views
  - Just select one or more backups to be pruned and run “**load backup -p**” (or **Run Prune Server Cmd**)



3rd Party Restore (Domino 12.0.1)

In case we still have time..

Slides have most of the details

Domino 12.0.1 3rd Party Restore

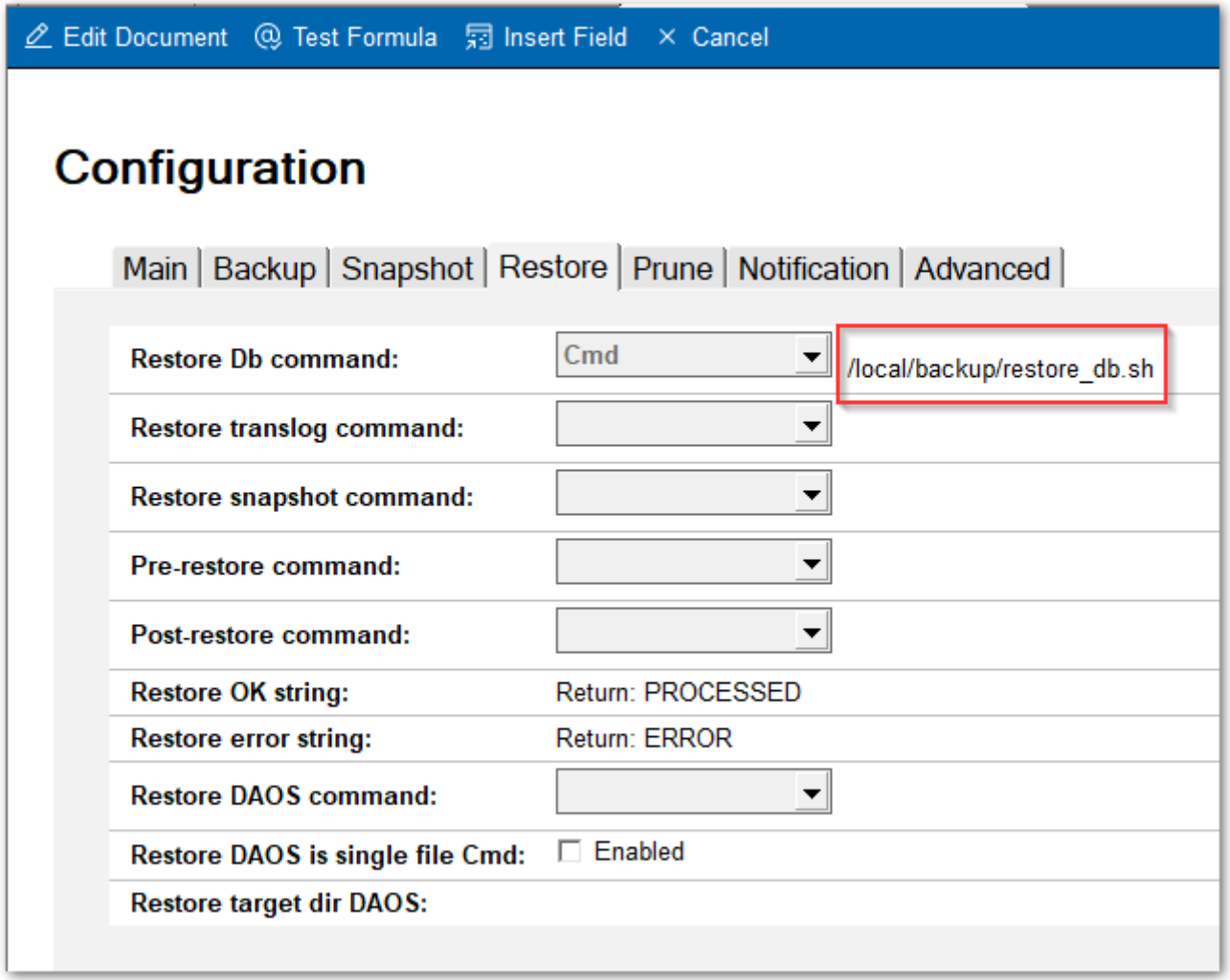
- Backup vendors with full Domino support leveraging the standard C-API interface can now leverage the Domino restore interface for restore operations

- **Restore 3rd party restore mode**
  - Allows to integrate with your favorite Domino aware backup solution
  - Leverages same type of integration used for the existing backup integrations

- Flow:
  - Specify database to restore & run restore operation
  - "**restore**" task executes integration script to trigger 3rd party restore operation
  - 3rd party restores database and brings it online
  - Domino **restore** performs additional restore operations (changing replica-ID, restoring documents & folders!)
  - Existing DAOS restore operations can be combined

Configure Domino 3rd Party Restore

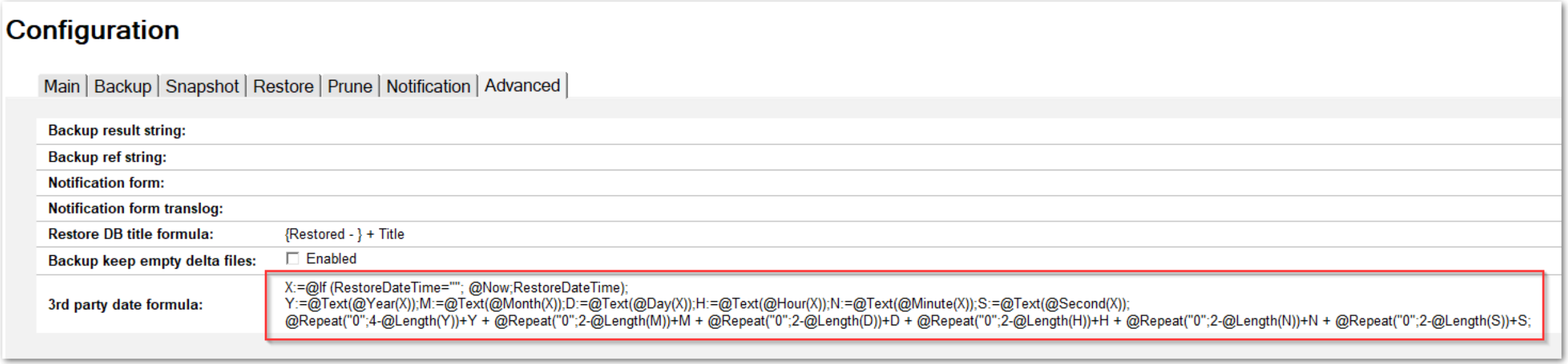
- Same type of integration you know from full integration operations
- Restore operation triggers the external backup vendor for a full restore including bringing the database online
- Domino restore operations will be executed immediately after restore completes
- Tip:
  - Restore time cannot be passed via **Cmd** operation
  - Best choice is probably “**Cmd Formula**”
  - Allows to pass any field in restore document



Configure Domino 3rd Party Restore

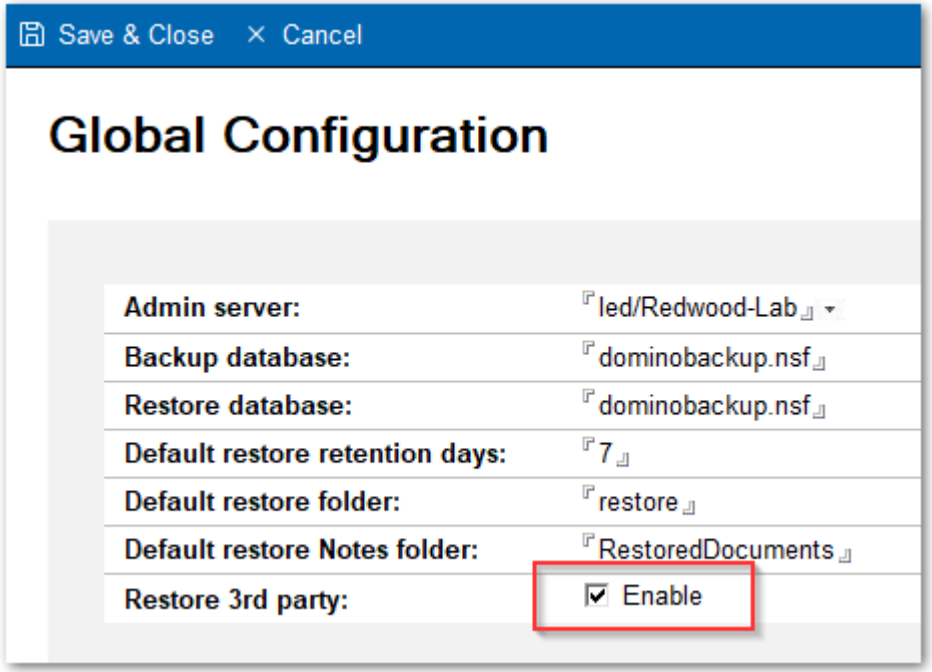
- Tip, if you only want to change the date, there is a 3rd party date formula

```
- X:=@If (RestoreDateTime=""; @Now;RestoreDateTime);
  Y:=@Text(@Year(X));M:=@Text(@Month(X));D:=@Text(@Day(X));H:=@Text(@Hour(X));N:=@Text(@Min
  ute(X));S:=@Text(@Second(X));
  @Repeat("0";4-@Length(Y))+Y + @Repeat("0";2-@Length(M))+M + @Repeat("0";2-@Length(D))+D +
  @Repeat("0";2-@Length(H))+H + @Repeat("0";2-@Length(N))+N + @Repeat("0";2-@Length(S))+S;
```



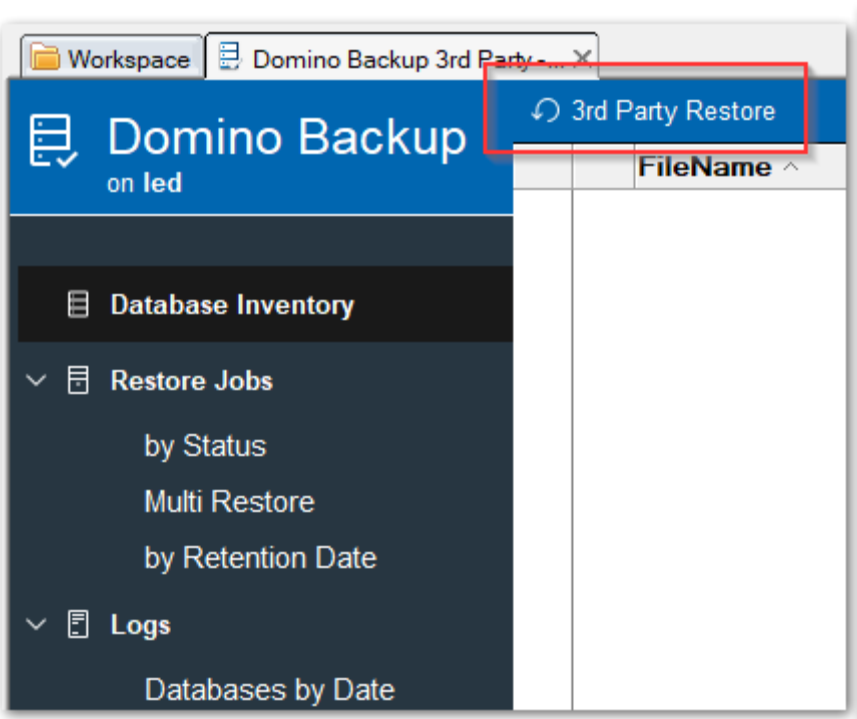
Enable 3rd Party Restore Operations

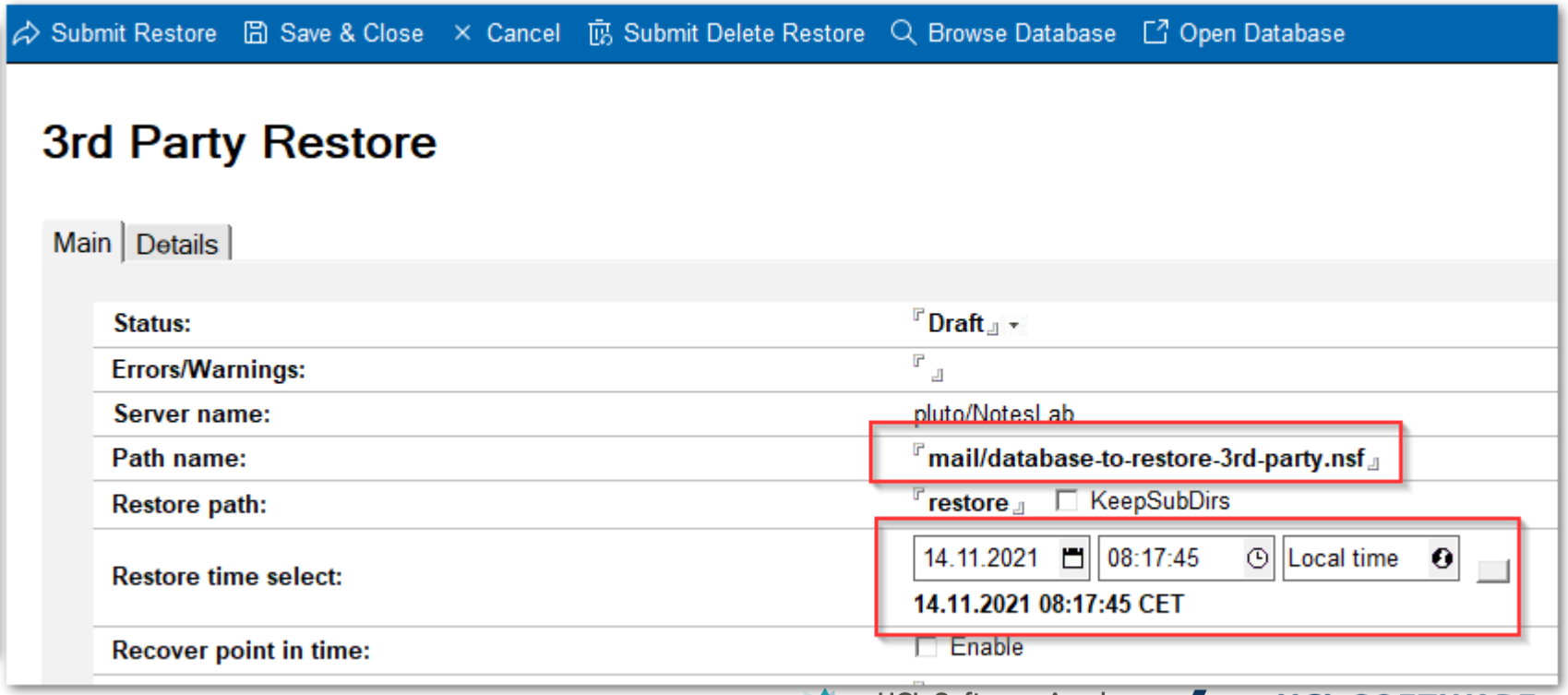
- Open global configuration
- Enable “**Restore 3rd party**”
- Brings **dominobackup.nsf** into 3rd party operations mode



Perform 3rd Party Restore

- 3rd party restore can obviously not use the database inventory
- Restore operations are triggered by entering the database name manually





Tip: 3rd Party Restore → Select Databases via SmartIcon

- SmartIcons are globally available in any database
- **Idea:** Run SmartIcon on **person/mail-in** or **catalog.nsf** document to create a restore document

```
Server := @Subset(@DbName; 1);
BackupDB := "dominobackup.nsf";
@If (Server = ""; @Prompt([Ok];"Error"; "Cannot run on local database!"); "");
@If (Server = ""; @Return (""); "");
Database := @If (Mailfile != "";
@If (@Ends(@LowerCase(MailFile);".nsf");MailFile;MailFile+".nsf"); PathName !="";PathName;"");
@Command([Compose]; Server : BackupDB; "RestoreJob");
@UpdateFormulaContext;
@SetField ("ServerName"; Server);
@SetField ("PathnameRestore"; Database);
@Command([EditGotoField]; "RestorePath");
```

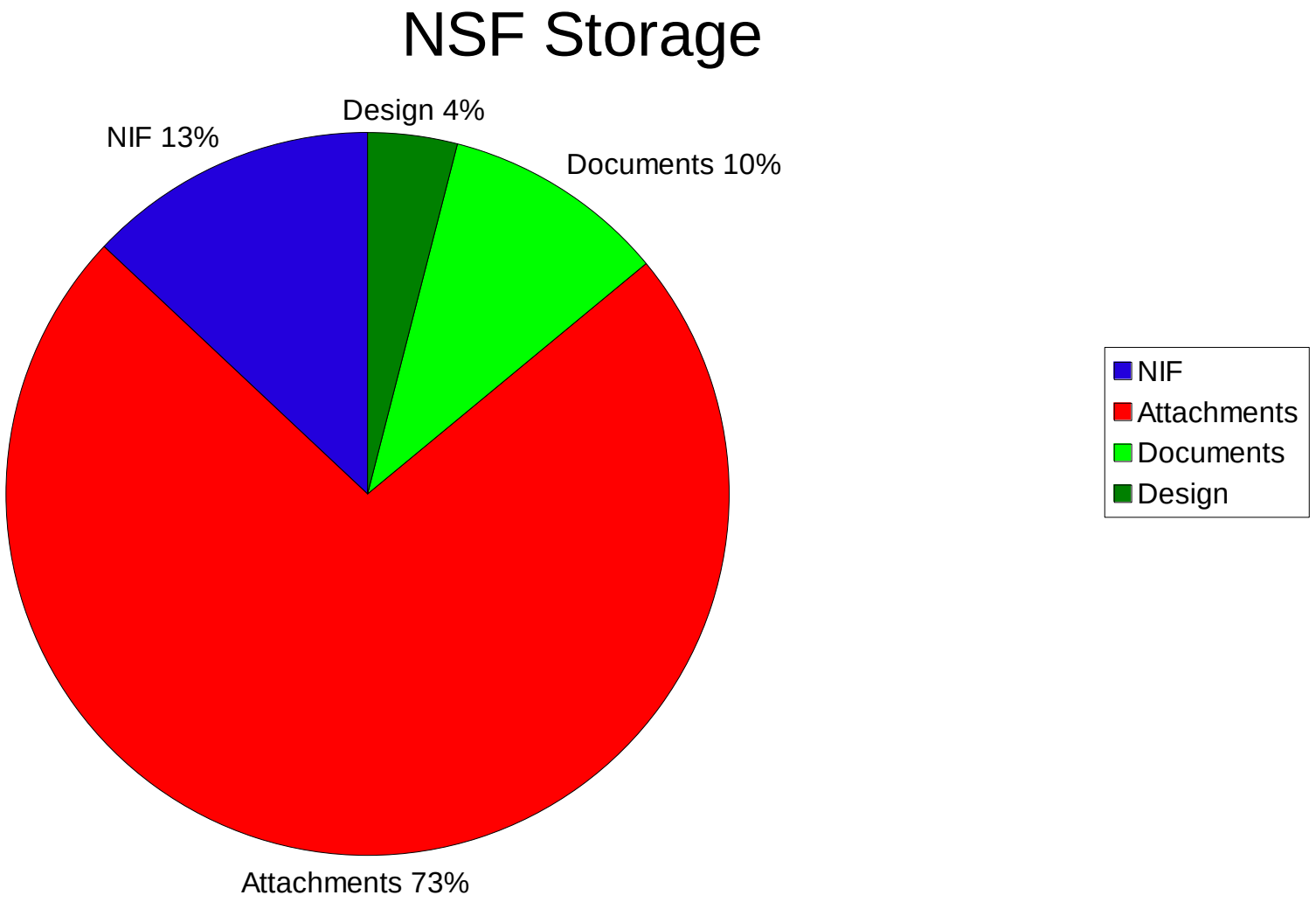
Q&A

Domino Storage Optimization

Domino Storage Optimization before Backup!

- Standard storage is cheap
  - **I/O intensive storage is more expensive**
  - High available storage can be more expensive as well
  - **Storage you need to backup every night can be expensive for your company over time!**
  - **The night has usually only ~12 hours for backup..**
- Domino provides many storage optimization options
  - Most of them available since Domino 8.5.x!
- You have to revisit your current database maintenance operations!
  - Database maintenance should usually scheduled before the backup
  - You don't need to compact nor fixup databases every night!
- **Domino storage optimization and database maintenance operations have to be addressed before you can optimize your backup!**

NSF Database Size before Optimization



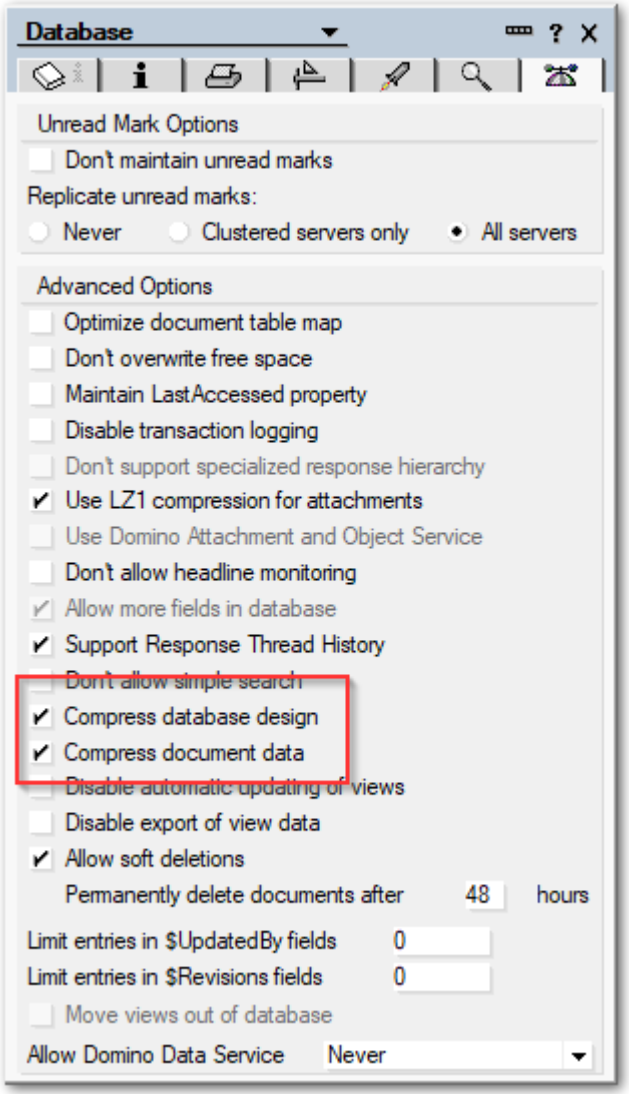
Design Note Compression (Domino 8.0)

- Design is stored **compressed** on disk
  - Only internal storage format changes → No difference visible to applications even on API layer
  - Opening the design notes automatically uncompressed the design
    - Tip: Enable network compression to reduce network I/O for remote operations

- Introduced in ODS 48 and enabled for Domino templates by default
- Reduces design of database design by **40 – 50%**
- Just marginal storage reduction on it's own → but also lowers I/O for design by 50%
- Example Mail 12.0.1 template:
  - Uncompressed **37 MB**
  - Compressed (default): **17 MB**

Document Compression (Domino 8.0.1)

- Works in the same way as design compression just for documents
  - All items including richtext
  - Only internal storage format changed. Also transparent to all APIs
  - Network compression helps for remote transactions
- **Reduces the size of a document by 40-50%**
  - Reduces storage and reduces I/O load!
  - **Needs less backup space & time every night!**
- Enabling design and document compression
  - Load **compact -n -v -*** → enables database properties
    - Compact is required to re-compress existing design/data
  - load **DBMT –ct 4**



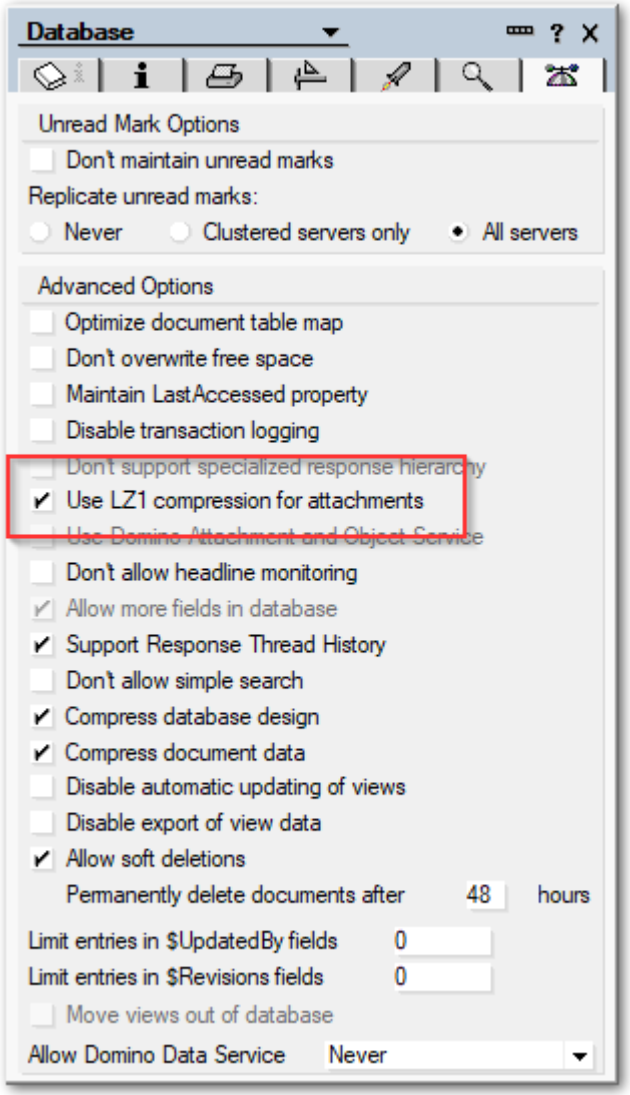
Design and Document compression I/O Reduction

- Main focus is I/O reduction
  - CPU utilization stays the same
  - Amount of I/O and even more important: **IOPS** (I/O per second) is reduced by 20%
- Data from a Lotusphere 2008 presentation

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

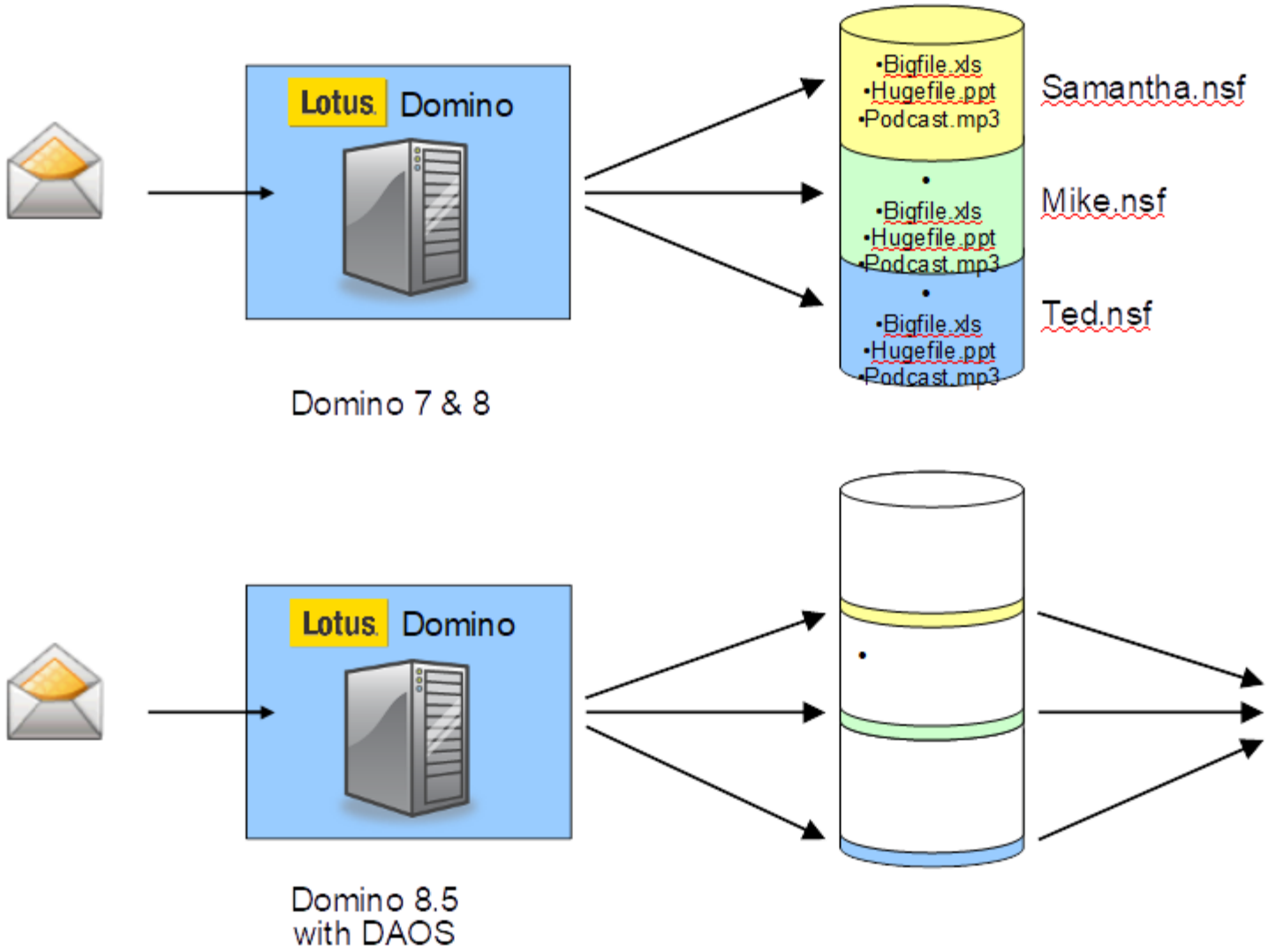
Attachment Compression

- Compresses attachments with LZ1 automatically
- Only internal representation of attachment changes
  - Transparent to application
- Compression rate is checked before storing compressed
  - Helpful when attachment is already compressed
  - Existing compress formats including image compression are not compressed



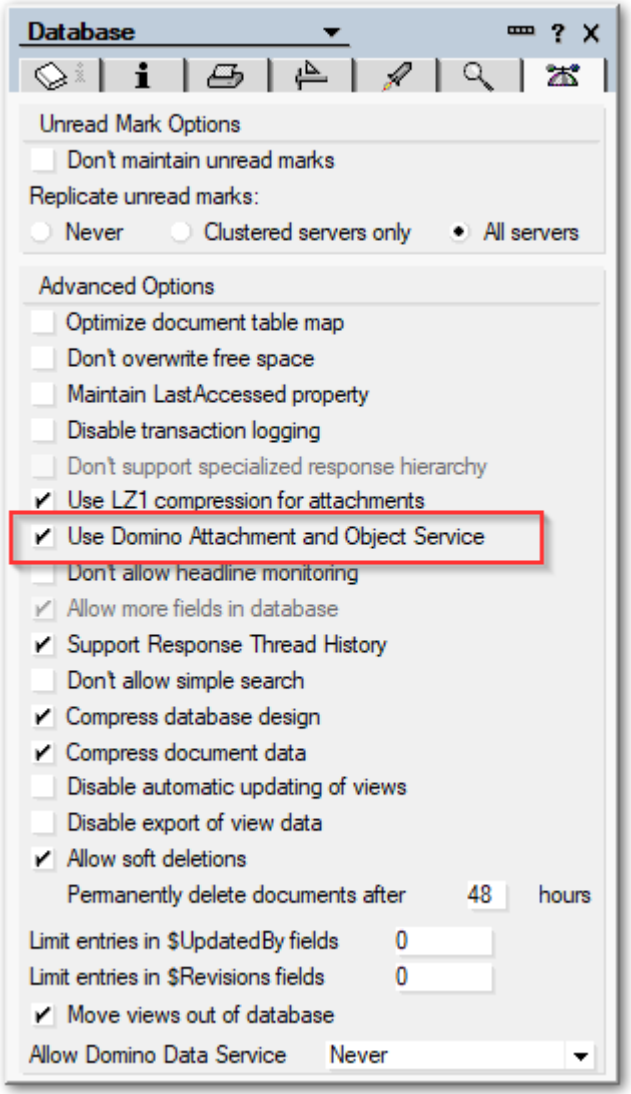
DAOS – Domino Attachment and Object Service

- Available since Domino 8.5
- Moves attachments to a separate file system with one *.NLO file per attachment
- Deduplicates attachments
- Reduces the NSF files up to 70%!
- Historic graphics from 8.5 used on purpose ;-)



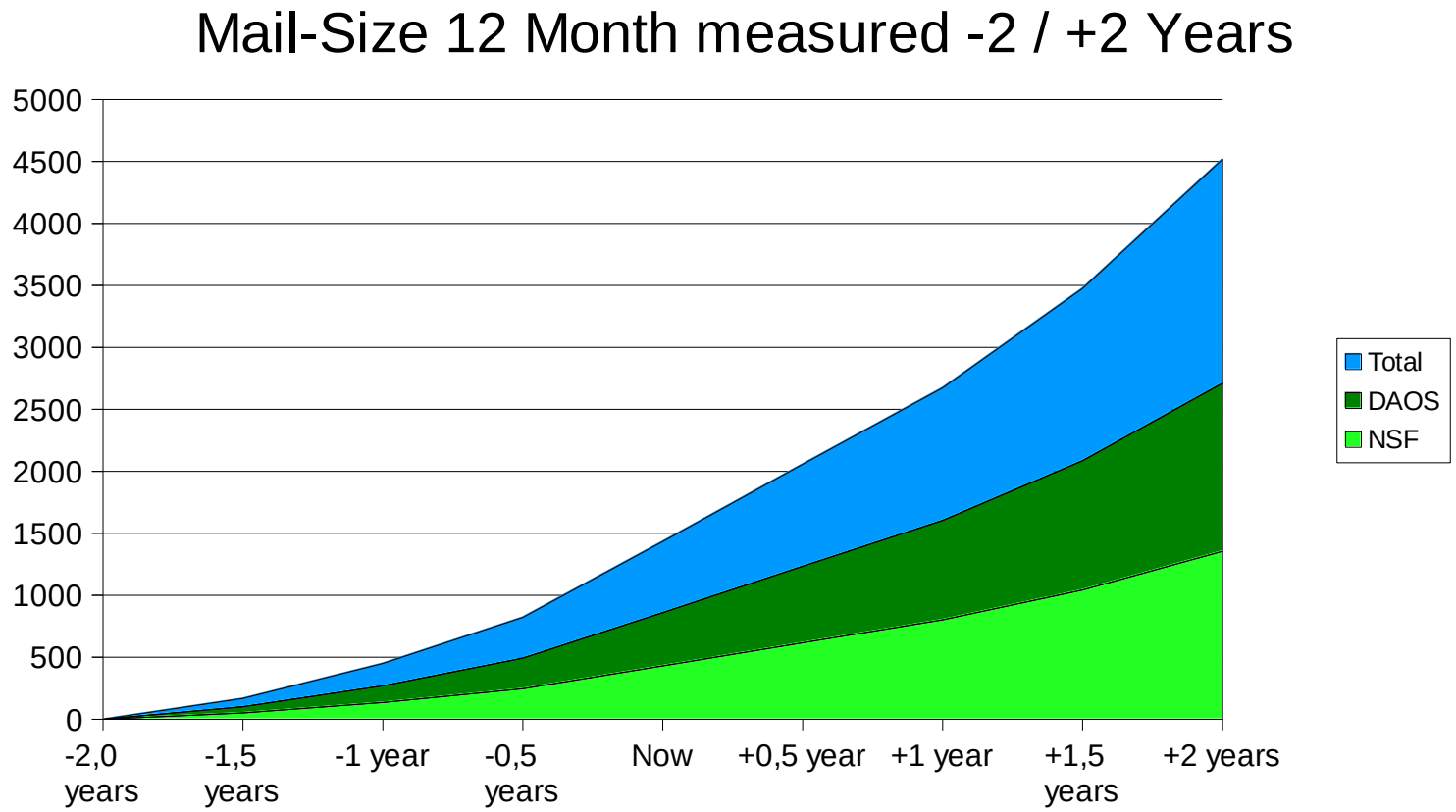
DAOS Benefits

- Reduce disk space depending on degree of content overlap
  - **Usually 30-50% storage reduction by deduplication**
  - **NSF file is reduced up to 70%**
- Improved performance
  - **Attachments are written once per server**
  - Less IOPS, with larger block size
- Large objects are stored outside the NSF in static files
  - Improve compact performance (less data stored + no large objects moved)
  - Reduce file-system fragmentation
  - **Large reduction in incremental backup costs**
  - DAOS repository isolates large blocks of data into separate, unchanging files
- Clients check if DAOS object is already on server before sending!



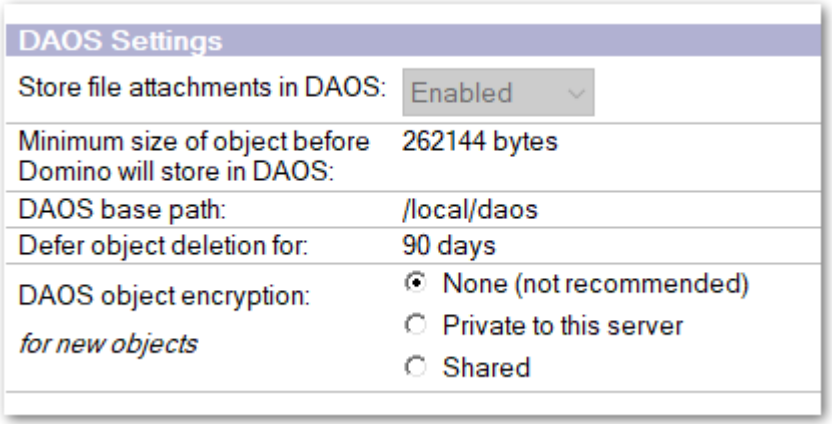
Mail Database Size with DAOS

- 40-50% total space reduction by deduplication
- Reduction of NSF by 60 – 70%
- Almost linear growth for NSF size with DAOS enabled



Enable DAOS

- Configure DAOS store on separate disk (server document)
  - On Linux and AIX it should be a sub directory of a mount point
- Specify DAOS threshold: 128 KB - 512 KB
  - Usually 256 K is a good starting point
- Disable DAOS encryption
  - before Domino 12.x via Notes.ini: **DAOS_ENCRYPT_NLO=0**
- Lower number of files per sub directory from 40000 to 10000
  - Notes:ini **DAOS_MAX_FILES_PER_SUBCONTAINER=10000**
- Wait for settings to be stored in notes.ini and restart server



- **Load compact –DAOS ON mail/**

  to enable DAOS settings
- **Load DBMT mail/**

  to move attachments to DAOS

NIFNSF

- NIF = **N**otes **I**ndex **F**acility
- View/Folder index data is stored in NSF file by default
- NIFNSF allows to store NIF data in a separate file
  - Available since Domino 9.0.1 FP8
- **For mail databases NIF can be up to 10% of the total space used**
  - For applications the index can be even bigger
- Only NSF files require backup. New *.NDX files only contain index data
- Separate file allows optimized locking of file
  - Important for runtime and for indexing!

Configure NIFNSF

- **NIFNSFENABLE=1**
  - Enables NIFNSF in general
  - By default NDX file is stored next to NSF file
- **NIFBASEPATH=/local/nif (Optional)**
  - Use different base path outside the data directory for NIF data
- **CREATE_NIFNSF_DATABASES=1 (Optional)**
  - Create new databases with NIFNSF enabled

- Enable per database:
  - **Load compact –NIFNSF ON**
- Move existing index data to NDX file
  - **Load DBMT**

Move FT Index to different disk

- Specially for snapshot backups, moving FT to a different disk/volume is important
- Notes.ini setting to define a directory outside notesdata directory.
- Example on Linux: **FTBASEPATH=/local/ft**
- Allows to store FT index into any location outside data directory → separate disk
  - It should be a local disk, not a remote mount!
  - Same I/O requirements than NSF

DBMT replaces Compact & Fixup

- Available since Domino 8.5, but still not used by many admins
- Allows fixup, updall and compact operations in one task
- Multi threaded and can run multiple compacts in parallel
- Copy style backup with disk space pre-allocation when creating a new NSF
  - New database with low number of fragments
  - Notes.ini **DBMT_PREFORMAT_PERCENT=n** to specify how much space is pre-allocated
  - DAOS aware calculating the new size in advance – e.g. DBMT_PREFORMAT_PERCENT=110
- Improvement of maintenance operations for
  - Weekend compact operations
  - Daily NIF and NSF operations

DBMT Example

- During the week only maintain indexes and database purge etc
  - No compact, 4 FT and 4 NIF update threads in parallel. Rebuild FT index every 30 days

  dbmt -ut 4 -ft 4 -ct 0

- **Saturday before backup**
  - Compact 4 threads only if database has more than 60 MB free space for max 3 hours
  - then compact all databases not compacted for 21 days for max 3 hours

  dbmt -ut 0 -ft 4 -ct 4 -tl 180 -ncl -S 60M –cnd 1

  dbmt -ut 0 -ft 4 -ct 4 -tl 180 -ncl -cnd 21

- **Sunday**
  - Update NIF index and FT index with 4 threads each and rebuild FT Index if 30 days old

  dbmt -ut 4 -ft 4 -ct 0 -fnd 30

Storage Optimization Summary

- **Leverage DAOS!**
  - Reduces NSF files up to **~70%**
  - DAOS is a simple file-backup of static files!
  - Recommended threshold: **~256 KB**
- **Use database design and document compression**
  - Saves **~50%** of design/document data
- **Enable NIFNSF to store index outside the NSF**
  - Reduces NSF size by **~10%** of the original database
- **Use DBMT to maintain your databases**
  - Don’t compact too often!

Leverage Domino archiving

- Database size grows exponentially
  - Usually doubles every 12 to 18 month
- Archiving lowers the exponential growth factor of your active databases!
- Archiving splits databases into
  - Smaller active often changing database
  - Larger static databases, with less I/O requirements
- Tip: If you archive once per month, you only need to backup only once after archiving
- Domino server based archiving is policy based and easy to setup