

Dissecting IBM Notes Documents: Hidden Secrets and Tricks Unveiled

Ben Menesi

Speaker

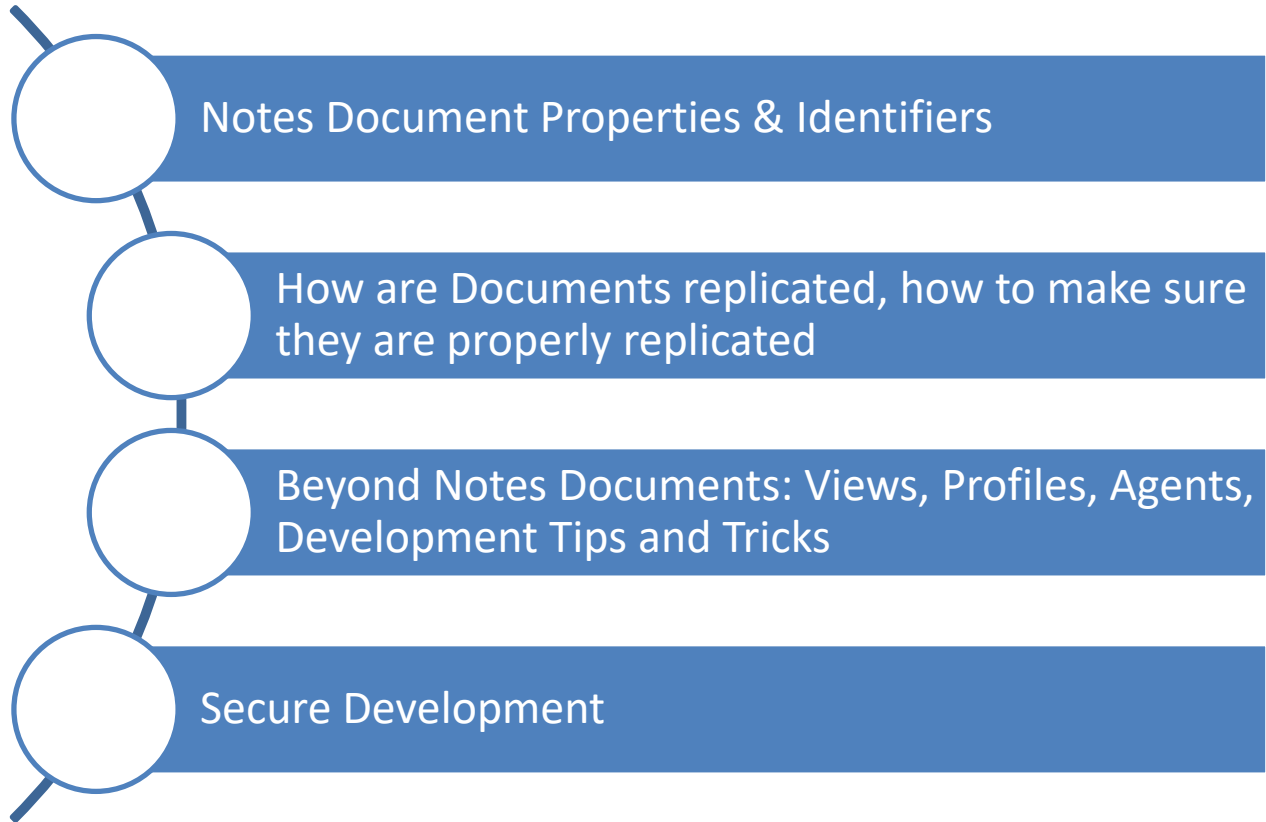
- Head of Product at Ytria
- IBM Notes Admin & Developer for various enterprise customers
- IBM Certified Advanced System Admin & Developer, Advanced Security Professional ND6+
- Wearing both Admin and Dev. Hats
- Speaker at IBM Connect, Engage, Admincamp etc...
- IBM Champion 2014 – 2017



@BenMenesi

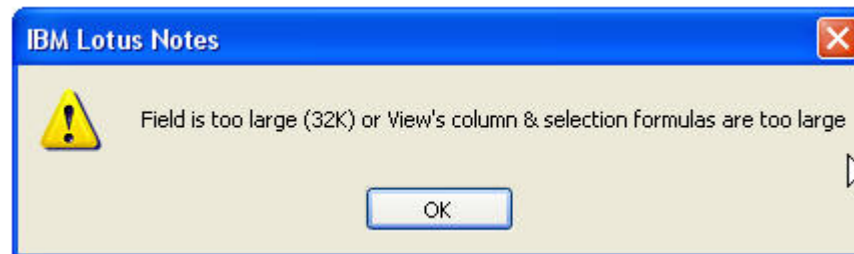
<http://www.ytria.com>

Agenda



Document Properties

- Summary
 - Flag for document items that allow the item to be displayed in views
 - Watch out! There is two different limitations here!
 - 1.) An item that's part of the summary, can't be larger than 32k in size
 - 2.) The summary of all Summary item sizes can't be larger than 64k
 - Or else we get



- New: with FP9 / FP10, both limitations are increased to 64MB

Document Identifiers

- Document Universal ID
 - 16-byte identifier that uniquely identifies a document across replicas
 - @DocumentUniqueID in formulas, NotesDocument.UniversalID in LS
- Note ID
 - Uniquely identifies documents in ONE notes database
 - @NoteID in formulas, NotesDocument.NoteID in LS
 - Much smaller in size than UNIDs

The screenshot shows the 'Note Info' tab of a document's properties. The UNID field is highlighted with an orange circle and contains the value '0EAEAE7CFA9231DD852573D60076A605'. The NoteID field is also highlighted with an orange circle and contains the value '7658 (h1DEA)'. Other fields include Class (Document (0x0001)), Created by (Mister Admin/Demo2), Modified by (John Admin/ACME), SeqNum (16), Nb Fields (81), Size (5.43 KB (5,559 bytes)), and various timestamps for initial creation and modification.

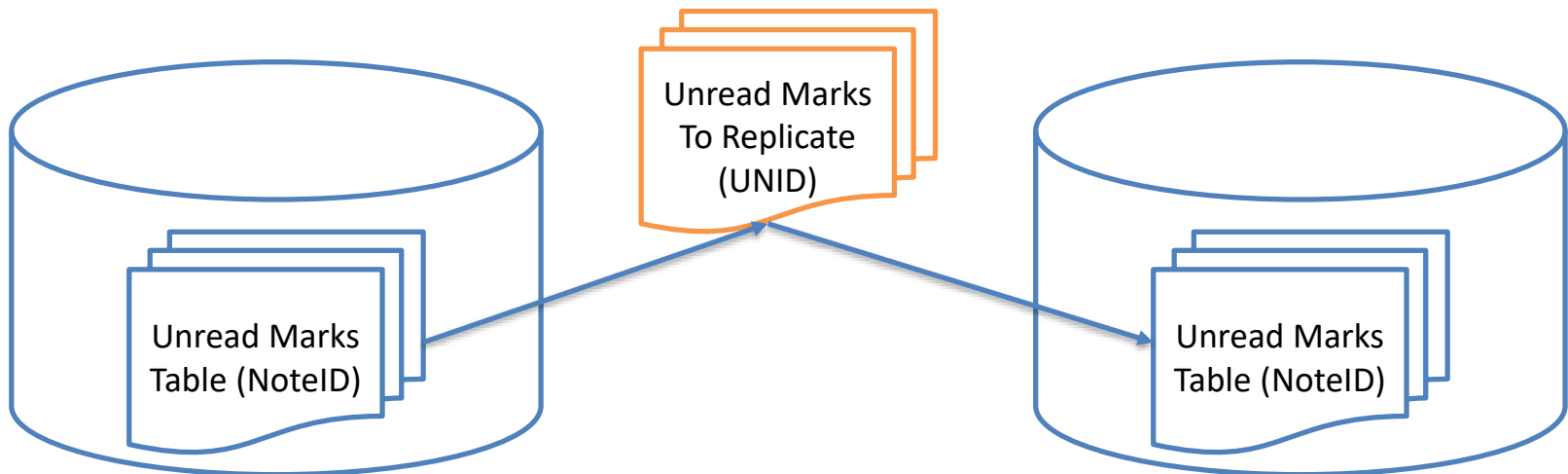
Field	Value
UNID	0EAEAE7CFA9231DD852573D60076A605
Class	Document (0x0001)
Created by	Mister Admin/Demo2
Modified by	John Admin/ACME
SeqNum	16
Nb Fields	81
Size	5.43 KB (5,559 bytes)
NoteID	7658 (h1DEA)
Flags	0x100 (0x100: -unknown-)
Created Initially	2008 01-20- 10:35 PM
Modified Initially	2016 09-06- 01:17 PM
Added In this file	2016 09-05- 05:36 PM
Modified In this file	2016 09-06- 01:17 PM
Accessed In this file	2016 09-06- 01:17 PM

Document Identifiers

- NotelDs – continued
 - Because they are smaller in size: Domino uses them for various reasons
 - NotelD Table: every single ‘note’ in a database
 - “Unable to extend Note ID table – Insufficient Memory” error
 - Usually due to a LOT of Deletion stubs
 - Manage Deletion stub cutoff date + enable PIRC (later)
 - Folder contents
 - It’s not the documents that ‘know’ which folder they belong to
 - Folders keep a notelD table for all documents that belong to them
 - Tip: Folder Reference Attribute is cool because it identifies what folders a document belongs to, but does NOT maintain folder membership!

Document Identifiers

- NoteIDs – continued
 - Unread Marks Table: each user has their own tables of unread NoteIDs
 - Issues with replication because:



Document Identifiers

- \$Revisions
 - Standard Document field that contains 8 byte entries for each time a doc has been saved.
 - CAUTION! There can be a DB level limit here
 - Good to know: NO \$Revisions for Design Elements
- \$Updatedby
 - Stores a list of editors (Canonical names = > 40 bytes). First entry is usually creator, but if DB limit hit, won't be accurate.
 - Computed when composed created by field is advised.
 - BEWARE! Usually not in sync with \$Revisions!
 - If there are multiple modifications by the same person after each other, it'll only record ONE entry.

Document Identifiers

- \$UpdatedBy item: No duplicate entries when same person saves document multiple times in a row (space saving considerations)

The image shows two screenshots of the IBM Notes interface. The left screenshot displays the \$Revisions field, which is a Date list. The right screenshot displays the \$UpdatedBy field, which is a Names field. Orange arrows indicate the mapping between the entries in the \$UpdatedBy field and the corresponding entries in the \$Revisions field.

Field	Value
\$Revisions	04/26/2013 12:52:40 PM
\$Revisions	04/26/2013 12:53:06 PM
\$Revisions	04/26/2013 12:53:58 PM
\$Revisions	04/26/2013 12:54:37 PM
\$UpdatedBy	CN=John Admin/O=ACME
\$UpdatedBy	CN=Dorothee Parker/O=ACME
\$UpdatedBy	CN=Joe Gowlan/O=ACME

Document Identifiers – Date Values

- Several Date Values for both documents and Design Elements

Creation Date & Time	Modification Date & Time	Deletion Date & Time
----------------------	--------------------------	----------------------

- But that's not all there is.

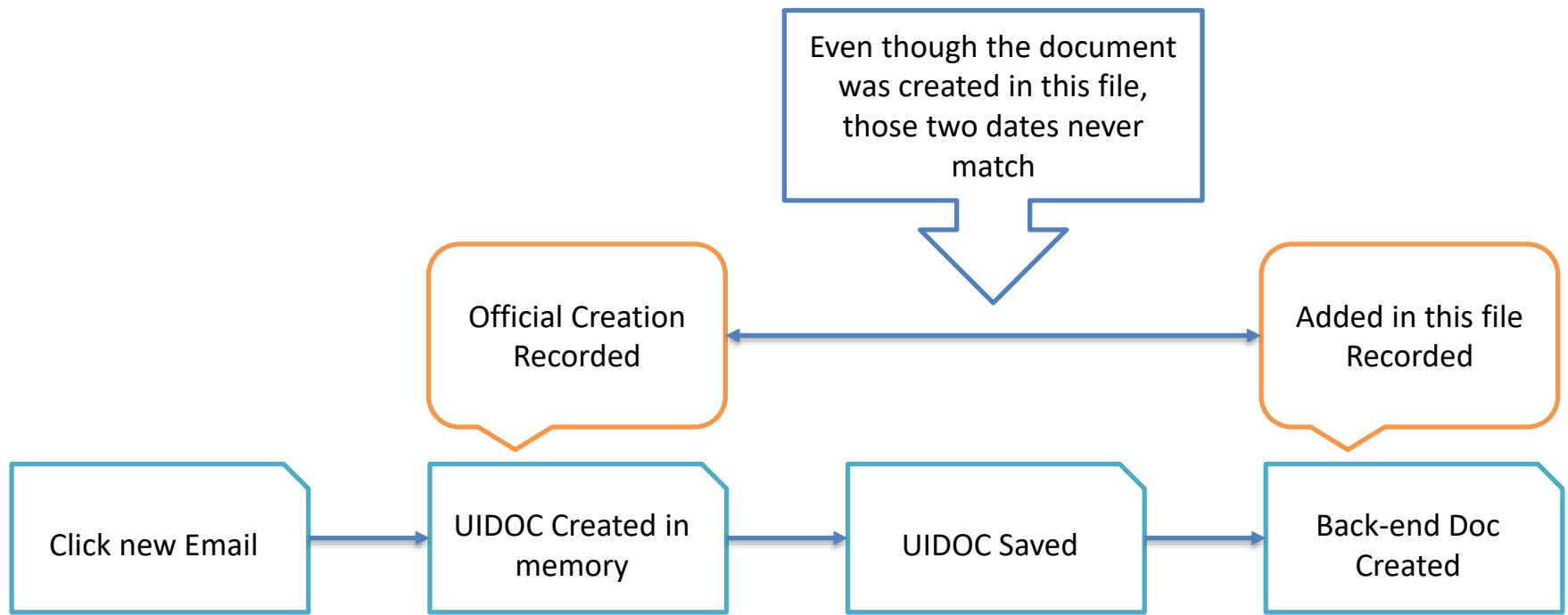
Creation Date & Time	Modification Date & Time	Deletion Date & Time
Official Creation	Initial Modification	Official Deletion
Added in This File	Modified In this File	Deleted in This File

Document Identifiers – Date Values

- Creation Dates
 - **Created Initially:** The ‘Birthday’ of your document / Design
 - To retrieve: @Created in formulas, NotesDocument.Created in LS
 - **Added to this file:** When the document / Design was replicated into this database
 - Previously only retrievable via the AddedToThisFile API Call
 - **NEW:** With Notes & Domino FP8, you can use the @Addedtothisfile formula!
 - Tip: You need FP8 on BOTH the server and the client
 - Why are these important? 2 Scenarios:
 - 1.) @Created = @AddedToThisFile: document was created in THIS replica
 - 2.) @Created < @AddedToThisFile: document was created OUTSIDE this replica, and time it took to replicate = @Created - @AddedToThisFile
 - ... This seems logical but is it actually true?

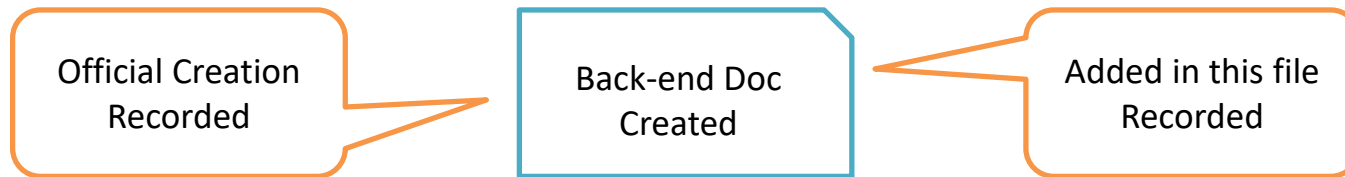
Document Identifiers – Date Values

- Creation Dates (Continued)
 - We experimented with FP8 and found that in the following scenario:



Document Identifiers – Date Values

- Creation Dates (Continued)
 - One exception: creating a document without using the UI (scanEZ or LS):

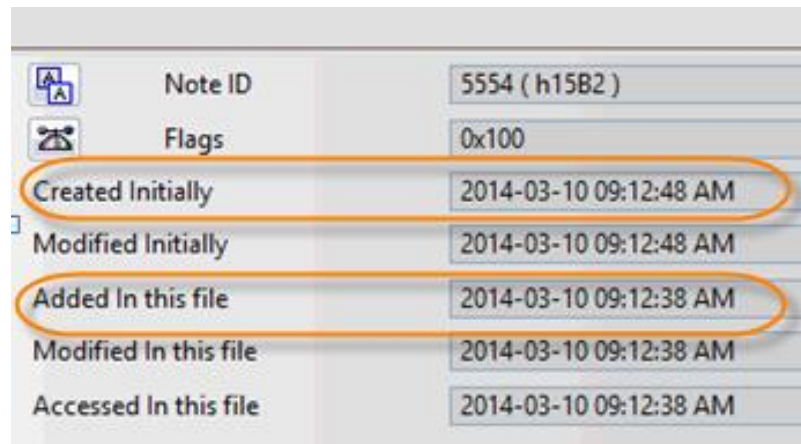


The screenshot shows a document's metadata and a formula editor. The formula editor contains the formula `@Created=@Addedtothisfile` and shows a resulting value of 1. The metadata table on the right lists various document properties and their values.

NotelD	4414 (h113E)
Flags	0x800 (0x800: -unknown-)
Created Initially	03/28/2017 04:41:56 AM
Modified Initially	03/28/2017 04:44:56 AM
Added In this file	03/28/2017 04:41:56 AM
Modified In this file	03/28/2017 04:44:56 AM
Accessed In this file	03/28/2017 04:44:56 AM

Document Identifiers – Date Values

- Creation Dates (Continued)
- Third scenario - Customer Example:
 - Added in this file BEFORE created initially



Note ID	5554 (h15B2)
Flags	0x100
Created Initially	2014-03-10 09:12:48 AM
Modified Initially	2014-03-10 09:12:48 AM
Added In this file	2014-03-10 09:12:38 AM
Modified In this file	2014-03-10 09:12:38 AM
Accessed In this file	2014-03-10 09:12:38 AM

- Tip: Ensure servers taking part in replication are getting their time from the same server!

Document Identifiers – Date Values

- Modification Dates
 - **Modified Initially:** Original last modification date of the document
 - To retrieve: @Modified in formulas, NotesDocument.Lastmodified in LS
 - **Modified in this file:** When the document / Design was modified in this particular replica
 - Previously only retrievable via the AddedToThisFile API Call
 - **NEW:** With Notes & Domino FP8, you can use the @Modifiedinthisfile formula!
 - Tip: You need FP8 on BOTH the server and the client
 - Why are these important?
 - We finally know what happened inside / outside our databases
 - Did you ever need to answer question like:
 - What happened in this database since this morning?
 - Where did these documents come from?

Document Identifiers – Date Values

- How can we answer these questions with a simple view in our applications?
- Create the following columns
 - Created Initially (using the @Created formula)
 - Added in this file (using the new @AddedToThisFile formula)
 - Modified Initially (using the @Modified formula)
 - Modified in this file (using the new @ModifiedInThisFile formula)
 - Modified By (using the formula @Subset(\$Updatedby;-1))
- What you get:

Created Initially ◊	Added In This File ◊	Modified Initially ◊	Modified In This File ◊	Modified By
01/22/2013 09:16:34 AM	01/22/2013 09:16:36 AM	02/22/2017 04:39:56 PM	02/22/2017 04:39:56 PM	John Admin
01/18/2005 12:31:28 PM	01/18/2005 12:32:27 PM	02/22/2017 04:39:56 PM	02/22/2017 04:39:56 PM	John Admin
01/18/2005 12:32:27 PM	08/30/2007 06:00:44 PM	02/22/2017 04:39:56 PM	02/22/2017 04:39:56 PM	John Admin
01/18/2005 12:32:38 PM	08/30/2007 06:00:44 PM	02/20/2017 09:05:54 PM	02/20/2017 09:05:54 PM	Benedek MENESI
01/18/2005 12:36:28 PM	01/18/2005 12:36:28 PM	02/20/2017 09:05:53 PM	02/20/2017 09:05:53 PM	Benedek MENESI
01/18/2005 12:36:28 PM	01/18/2005 12:36:28 PM	02/20/2017 09:05:54 PM	02/20/2017 09:05:54 PM	Benedek MENESI
01/18/2005 12:18:17 PM	01/18/2005 12:19:05 PM	02/22/2017 04:39:56 PM	02/22/2017 04:39:56 PM	John Admin
01/18/2005 12:19:05 PM	01/18/2005 12:19:05 PM	02/20/2017 09:05:53 PM	02/20/2017 09:05:53 PM	Benedek MENESI

Document Identifiers – Date Values

- We could go further with some additional columns but remember: @Created is recorded when the UI doc is created, while added to this file is the moment the back-end document is created in the NSF.
- This means:
 - @Created=@Addedtothisfile won't return 1 unless you use a workaround like:
 - @If((@Text(@AddedToThisFile)=@Text(@Created)):!@IsAvailable(\$Revisions);"Created in this replica";"Created outside")
- Also: It appears that @Modified and @Modifiedinthisfile are extremely sensitive. They return a Date & Time value which is shown to the seconds in the UI:

Document Identifiers – Date Values

- Modification Dates:
 - Use @Text() to get them to match up to the second

- So how to get around this in a view?

Formula Evaluation

Quick Lists

Basic Formulas: @Accessed Value Paste

Saved Formul... Delete Replace

View Formulas: Refresh Replace

Fields: \$Abstract Other... Paste

Values: asfas Paste

& | ! + - * / = <> < > <= >= Save Formula

@implode(@Text(@Modified):@Text(@Modifiedinthisfile); " = ")'+@newline+'But the result is...'+@Text(@modified=@Modifiedinthisfile)

|<-- <-- (37/37) --> --> Show History

Resulting Value:

03/28/2017 04:44:56 AM = 03/28/2017 04:44:56 AM
But the result is...0

Evaluate Close

Document Identifiers – Date Values

- Use the appropriate formulas to show the different creation & modification dates
- Use this formula to see if docs were created inside or outside:
 - `@If((@Text(@AddedToThisFile)=@Text(@Created)):!@IsAvailable($Revisions);"Created in this replica";"Created outside")`
- Use this formula to see if docs were modified inside or outside
 - `@Text(@Modified)=@Text(@Modifiedinthisfile)`
- And create a view like this:

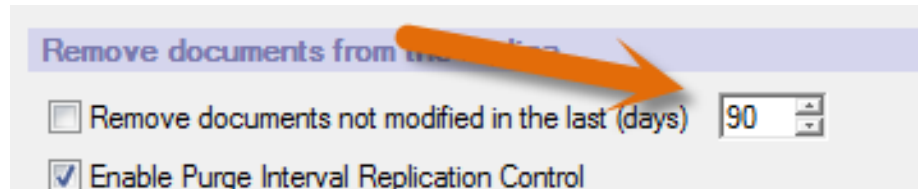
Created Initially ^	Added in This File ^	Created Location ^	Modified Initially ^	Modified In This File ^	Modified Location ^
03/28/2017	03/28/2017	✘ Created outside	03/28/2017 05:12:13 AM	03/28/2017 05:12:13 AM	✔ Modified in this replica
03/28/2017	03/28/2017	✔ Created in this replica	03/28/2017 05:11:30 AM	03/28/2017 05:11:50 AM	✘ Modified outside
03/28/2017	03/28/2017	✘ Created outside	03/28/2017 05:12:16 AM	03/28/2017 05:12:16 AM	✔ Modified in this replica
03/28/2017	03/28/2017	✔ Created in this replica	03/28/2017 05:11:04 AM	03/28/2017 05:11:50 AM	✘ Modified outside
03/28/2017	03/28/2017	✘ Created outside	03/28/2017 05:11:10 AM	03/28/2017 05:11:50 AM	✘ Modified outside

Document Identifiers – Date Values

- Deletion Stubs – what are they?
 - When a document is deleted from the database, a light place holder is created that contains little information
 - In case enabled, Soft Deletions are an interim step in this process
 - Soft Deletions have a different class, and are only different from deletion stubs because they actually do contain all their items
- What do deletion stubs contain?
 - Document UNID
 - Creation Time (Initial)
 - Deletion Time (Both initial & in this file)
 - Sequence number
- Deletion Dates
 - **Official Deletion:** When the deletion stub was first created
 - **Deleted in this file:** When the deletion stub was created in this file

Document Identifiers – Date Values

- Deletion Stubs – how long are they kept in the database?
 - By default they will be deleted after 90 days
 - Can be changed on Space Savers tab of replication settings window. (Not very obvious, but field value takes affect even if removal is not checked)



- Note: Purging of deletion stubs interval will be the 1/3rd of this value!
 - In the default case this means it'll happen every 30 days, which will possibly leave you with deletion stubs that are kept in the database for <120 days!

Document Identifiers – Date Values

- Customer Example
 - 2 servers in a cluster
 - Replica of mail files on secondary server
 - Quotas being blown on secondary server when users fail over
 - Same number of documents
 - Only difference found was number of deletion stubs
 - Conclusion: Purge only happens on database open
 - Replica on secondary server contained thousands more deletion stubs than replica on primary server

Document Identifiers – Date Values

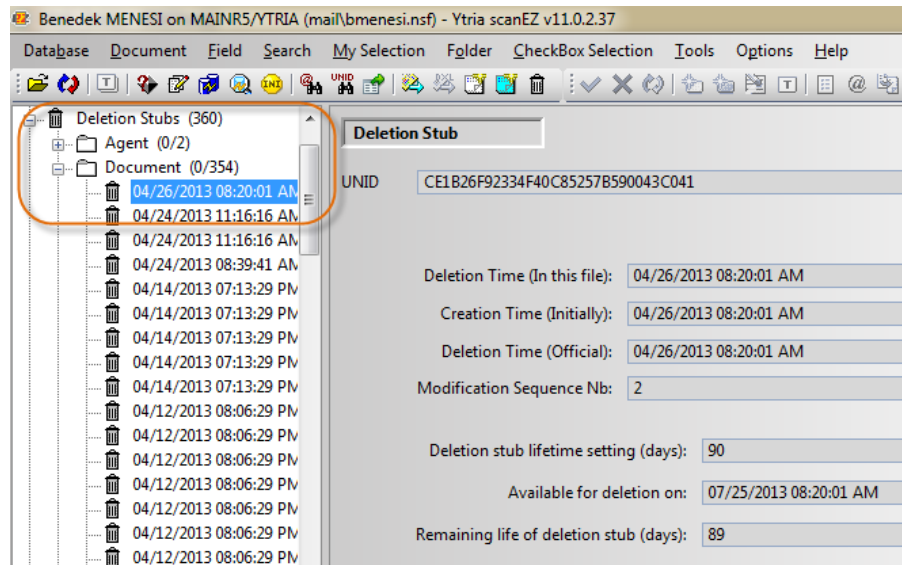
- Deletion Stubs - How to locate them?
 - NOT displayed in views (in fact, they aren't even documents – different class)
- TIP: When using GetdocumentbyUNID in LS, object returned by a Deletion Stub will raise an error when trying to work with it
 - doc is nothing is NOT enough protection here!
 - When getting document object, once “Is Nothing” check has been done, you should do an “ If TestDoc.Size = 0 Then” type check to handle any errors due to deletion stubs.
 - Favorite comment below an article regarding this:

COMMENTS

1 - Holy crap. I have some code to go patch!

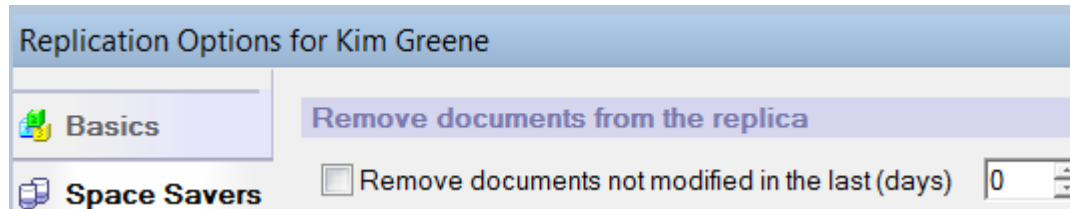
Document Identifiers – Date Values

- Deletion Stubs - How to locate them? (Continued)
 - Only way is to use the Notes API
 - Both NSFDbGetNoteInfo and NSFDbGetNoteInfoByUNID calls return the note header without attempting to open it, i.e. will let you gather the sequence number too
- Using scanEZ (lite / free version, too!)



Document Identifiers – Date Values

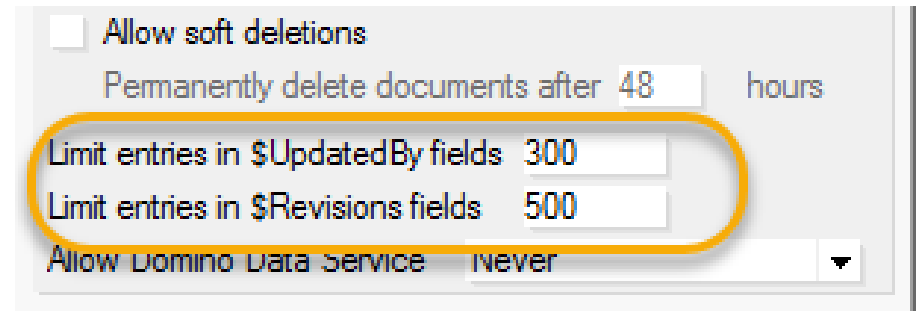
- Deletion Stubs - How to remove them?
 - Reset Purge interval
 - Unless the customer has a tool to get rid of deletion stubs manually (like scanEZ), changing the database replication settings -> space savers, and setting the parameter 'Remove documents not modified in the last (days)' to 0, closing and reopening the database clears the deletion stubs



- The TRICK, is to NOT put a checkmark next to 'Remove documents modified in the last (days)' as that will delete documents from the database that were not intended to be deleted!!!
- Use DataBase Maintenance Tool (DBMT) in Domino 9.x
- Clearing deletion stubs programmatically ONLY works through the C API

Sequence Numbers

- Sequence Number (used for versioning and consequently replication)
 - Document SQ Nr: Increases Every time a document is saved
 - Design Element SQ Number: Increases Every time design element is saved
 - Item level SQ Nr: Only increases IF item was modified during save operation!
- Ways to get it
 - @Elements(\$Revisions)
 - Watch out! Risky because there might be a limit here which in turn will return the wrong value
 - Using the Notes C API:
 - NSFNoteGetInfo
 - Part of the returned header



The screenshot shows a configuration dialog box with the following settings:

- Allow soft deletions
- Permanently delete documents after hours
- Limit entries in \$UpdatedBy fields
- Limit entries in \$Revisions fields
- Allow Domino Data Service

The two limit entries are circled in orange.

Sequence Numbers

- Item level Sequence Numbers (Customer Example)
 - Certain Documents modified, but item values reset overnight.
 - Screenshot from the night before:
 - Document SQ Number: 209, item sequence number: 209

The screenshot displays a database table with the following data:

Name	N.	Sequence Number	D.	Value
Path_Info	1	209	0	658
readers	1	209	0	656
Remote_User	1	209	0	8
Subject	1	209	0	659 This is a test document
ThreadId	1	209	0	7 BMEN-AKUBMT

Below the table, the document metadata form is shown with the following fields:

UNID	FF4784AF278987E8C12580F0002D15B7	NoteID	18558 (h487E)
Class	Document (0x0001)	Flags	0x0 (-not set-)
Created by	Benedek MENESI/YTRIA	Created Initially	2017 03-27- 10:12 AM
Modified by	Benedek MENESI/YTRIA	Modified Initially	2017 03-27- 10:23 AM
SeqNum	209	Added In this file	2017 03-27- 10:14 AM
Nb Fields	664	Modified In this file	2017 03-27- 10:23 AM

Sequence Numbers

- Item level Sequence Numbers (Customer Example - continued)
 - Screenshot from the morning after:
 - Document SQ Number: 289, item sequence number: 33

Name	N.	Sequence Number	D.	Value	
Path_Info	1	33	0	658	
readers	1	33	0	656	
Remote_User	1	33	0	8	
Subject	1	33	0	65	This is a test document
ThreadId	1	33	0	7	BMEN-AKUBMT

Rows Loaded: 664 | Rows Displayed: 664 | Rows Selected: 1

Field	Value
UNID	FF4784AF278987E8C12580F002D15B7
Class	Document (0x0001)
Created by	Benedek MENESI/YTRIA
Modified by	Benedek MENESI/YTRIA
SeqNum	289
Nb fields	664
NotelD	18558 (h487E)
Flags	0x100 (0x100: -unknown-)
Created Initially	2017 03-27- 10:12 AM
Modified Initially	2017 03-27- 10:26 AM
Added In this file	2017 03-27- 10:14 AM
Modified In this file	2017 03-27- 10:26 AM

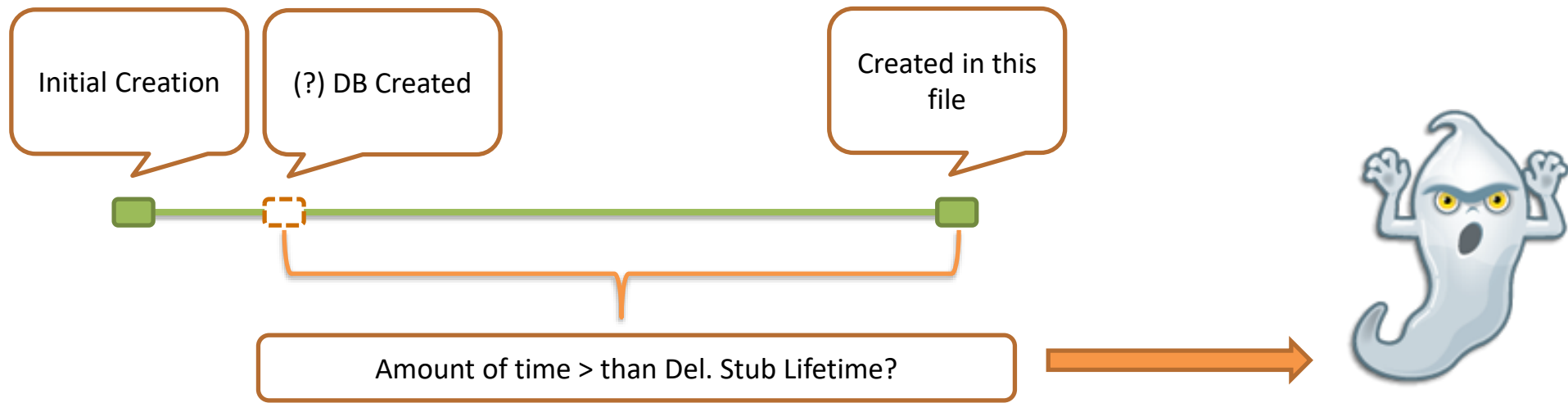
- Turns out, Item level Sequence Numbers reset after 256!

Document Identifiers – Date Values

- Another customer scenario: Ghost documents
 - You leave the office with an internationally replicating database that has 60k documents
 - You come to work in the morning, and all of a sudden, there are 75k documents
- What happened?
 - You have a replica somewhere (local OR server) which has not been replicating with your DB since more than the deletion stub lifetime setting
 - This means:
 - Documents have been deleted from the production version of the DB, but even their deletion stubs have been purged
 - These documents will be considered as NEW documents when replication happens

Document Identifiers – Date Values

- Use the Initial creation & created in this file dates
 - Initial creation @Created
 - Added in this file @Addedtothisfile
 - Difference between these two dates



Document Identifiers – Date Values

- With the new @Addedtothisfile formula you can build a view
- ... And categorize it by something like:
`@If(@Addedtothisfile-@Created > 90*24*60*60;150;””)`
- You get a view where you can identify ghost documents:

954 days, 3 hours, 47 minutes, 15 seconds	! Potential Resurrection
0 days, 0 hours, 1 minutes, 11 seconds	
0 days, 0 hours, 0 minutes, 37 seconds	
0 days, 0 hours, 0 minutes, 48 seconds	
0 days, 0 hours, 0 minutes, 0 seconds	
0 days, 0 hours, 0 minutes, 59 seconds	
0 days, 0 hours, 0 minutes, 0 seconds	
0 days, 0 hours, 0 minutes, 0 seconds	
954 days, 4 hours, 28 minutes, 17 seconds	! Potential Resurrection
954 days, 4 hours, 28 minutes, 6 seconds	! Potential Resurrection
0 days, 0 hours, 0 minutes, 2 seconds	

- Note: you'll need the DB Creation date to be sure, but there is no formula for this.

Document Identifiers – Date Values

- Preventive Actions – Ghosts
 - Use the PIRC (Purge Interval Replication Control) feature
 - If enabled, replication runs a background check to only let documents in when the difference between their initial creation and the time of replication is less than the Deletion stub lifetime setting
 - New with 8.5.3
 - If you create a new replica, replicate first and only then enable PIRC: otherwise old documents won't get replicated!

Remove documents from the replica

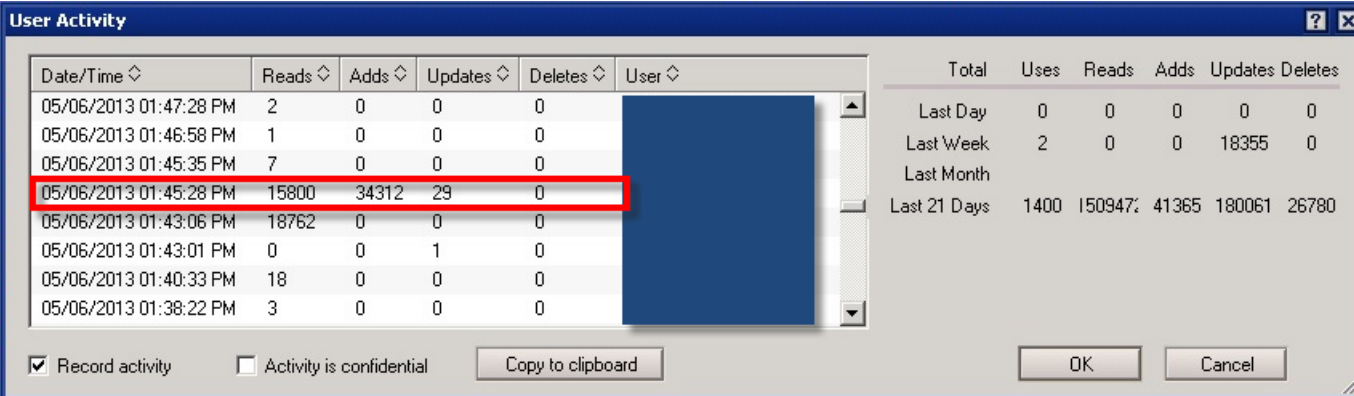
Remove documents not modified in the last (days)

Enable Purge Interval Replication Control

Prevent modified documents and deletions before [12/19/2012 02:12:43 PM] from replicating into database.

Document Identifiers – Date Values

- Customer Example
 - User reported in early May thousands (34,000+) documents deleted months prior reappeared
 - Application had been PIRC enabled
 - Server moved to new box
 - New replicas created via AdminP
 - BEWARE: PIRC database setting doesn't carry forward when creating new database copy



Date/Time	Reads	Adds	Updates	Deletes	User
05/06/2013 01:47:28 PM	2	0	0	0	
05/06/2013 01:46:58 PM	1	0	0	0	
05/06/2013 01:45:35 PM	7	0	0	0	
05/06/2013 01:45:28 PM	15800	34312	29	0	
05/06/2013 01:43:06 PM	18762	0	0	0	
05/06/2013 01:43:01 PM	0	0	1	0	
05/06/2013 01:40:33 PM	18	0	0	0	
05/06/2013 01:38:22 PM	3	0	0	0	

	Total	Uses	Reads	Adds	Updates	Deletes
Last Day	0	0	0	0	0	0
Last Week	2	0	0	18355	0	0
Last Month						
Last 21 Days	1400	150947	41365	180061	26780	

Document Identifiers – Date Values

- PIRC
 - Set notes.ini to show more information
 - DEBUG_REPL_PIRC
 - = 1 (for general info)
 - = 2 (for more details)
- Instruct the replicator server task to disregard PIRC
 - Replicate –NOPIRC
 - Will replicate the selected databases with no regards to the PIRC settings for the given session

Replication – How comparison happens

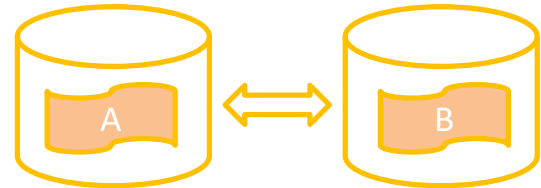
- Recap on how documents are compared
 - 0.) Need to have access to the database: whoever replicates needs ACL entries

Replicator	Cluster Replicator
1.) Replication History present?	1.) In memory streaming replication
2.) Establish list of documents <ul style="list-style-type: none">- Readers items important- Considers Selective formulas	2.) Ensures all in-memory changes are replicated <ul style="list-style-type: none">- Doesn't consider selective replication formulas
3.) Synchronizes changes	3.) Synchronizes changes

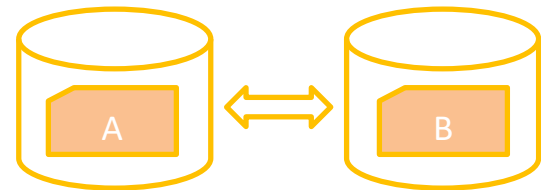
Replication – How comparison happens

- How are documents and designs compared?
 - Three major scenarios

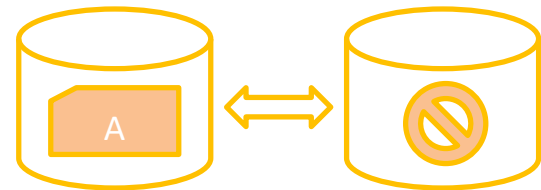
- Design Note (or Deletion Stub) exists in both databases



- Document / Profile document note exists in both databases

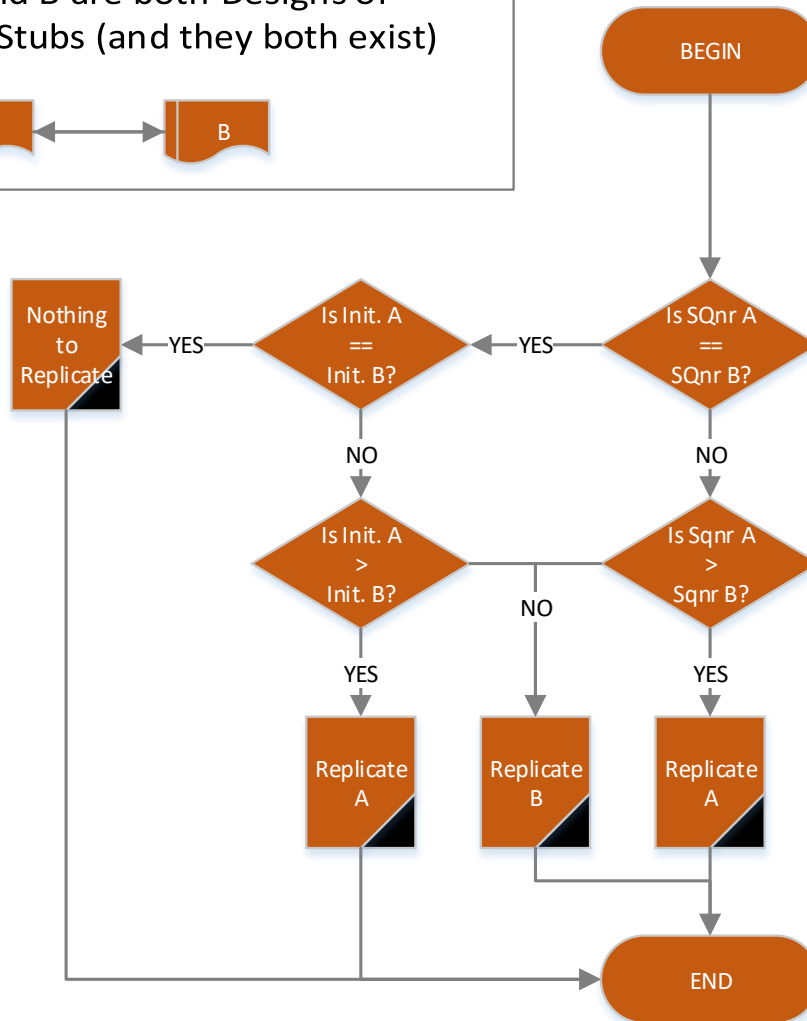


- Note doesn't exist in at least one of the databases (or is a deletion stub)

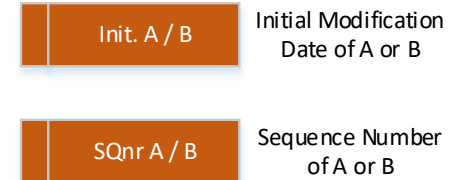


Replication – How comparison happens

1/3.) A and B are both Designs or Deletion Stubs (and they both exist)




Abbreviations





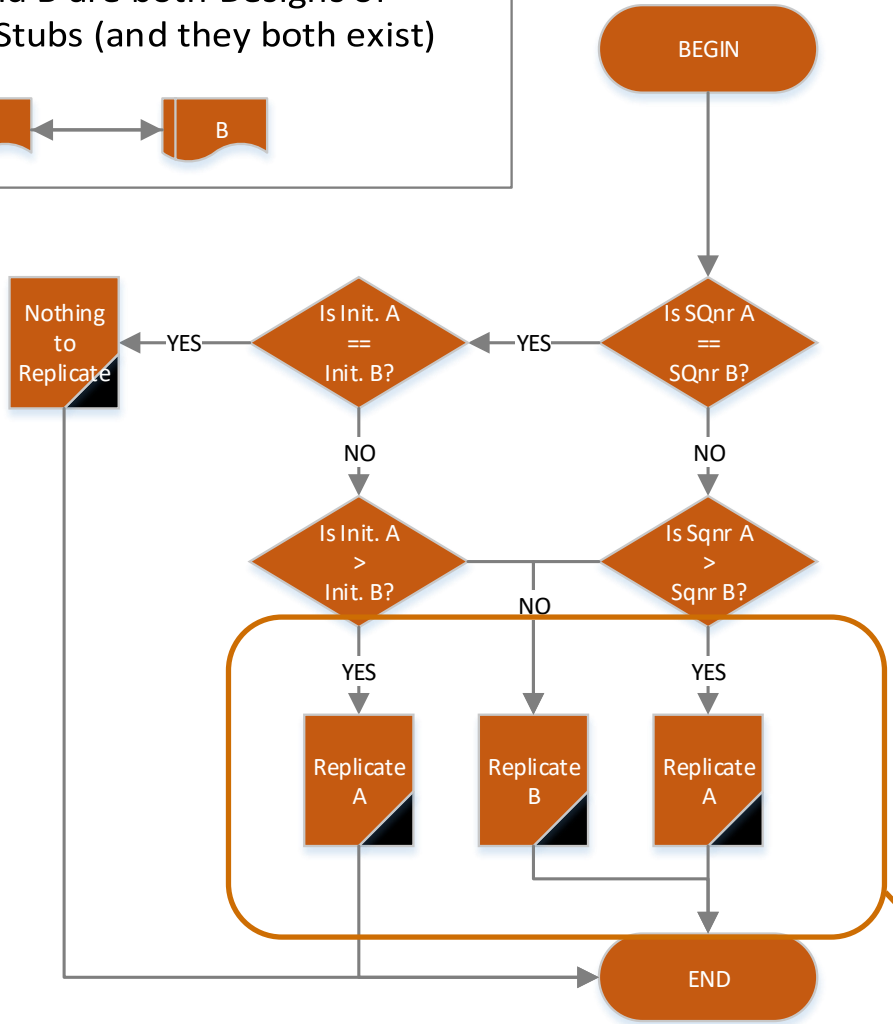
Replication – How comparison happens

1/3.) A and B are both Designs or Deletion Stubs (and they both exist)



Abbreviations

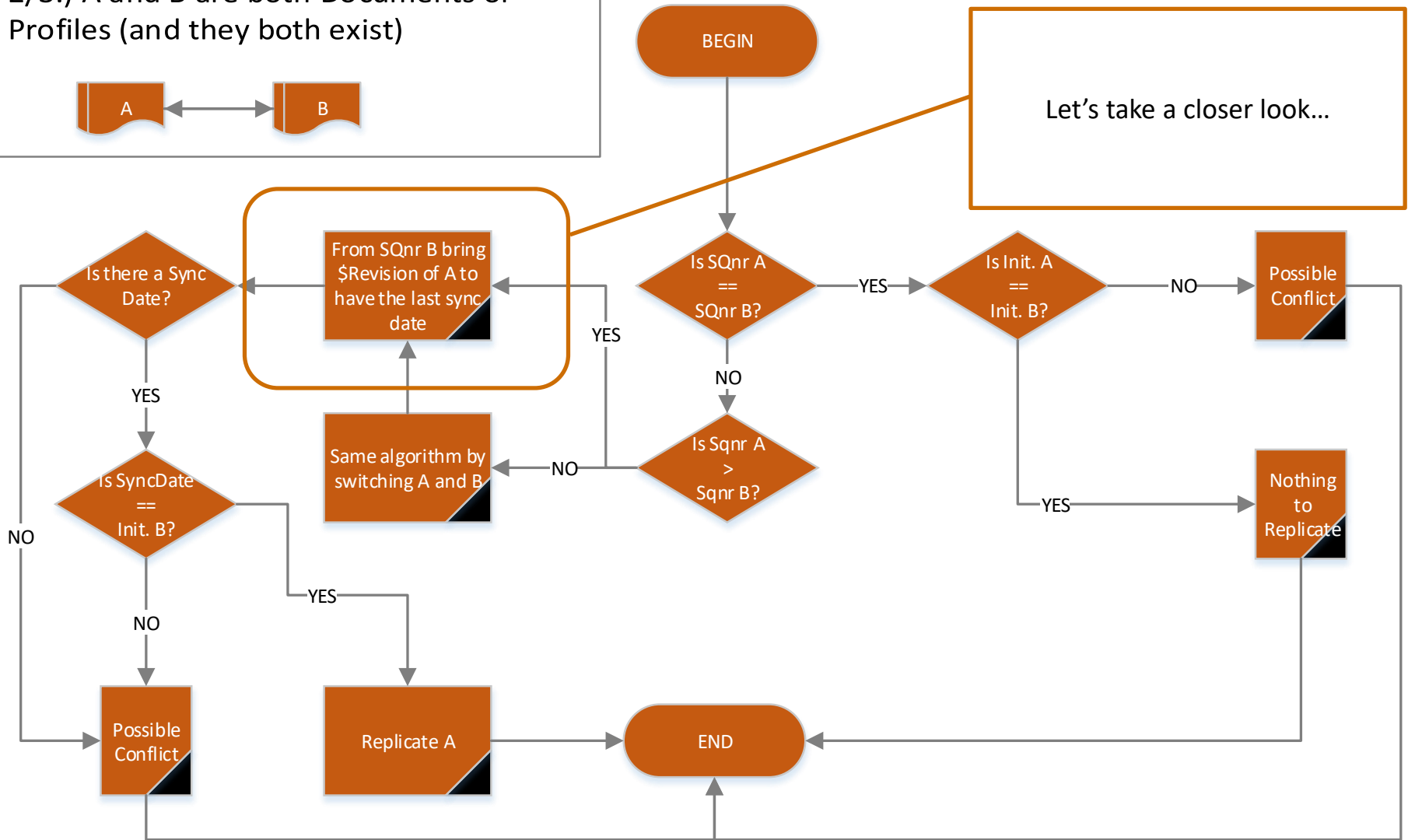
	Init. A / B	Initial Modification Date of A or B
	SQnr A / B	Sequence Number of A or B



TRICKY PART: What if both designs have been modified? No Conflict created, note with the bigger SQnr. Wins!

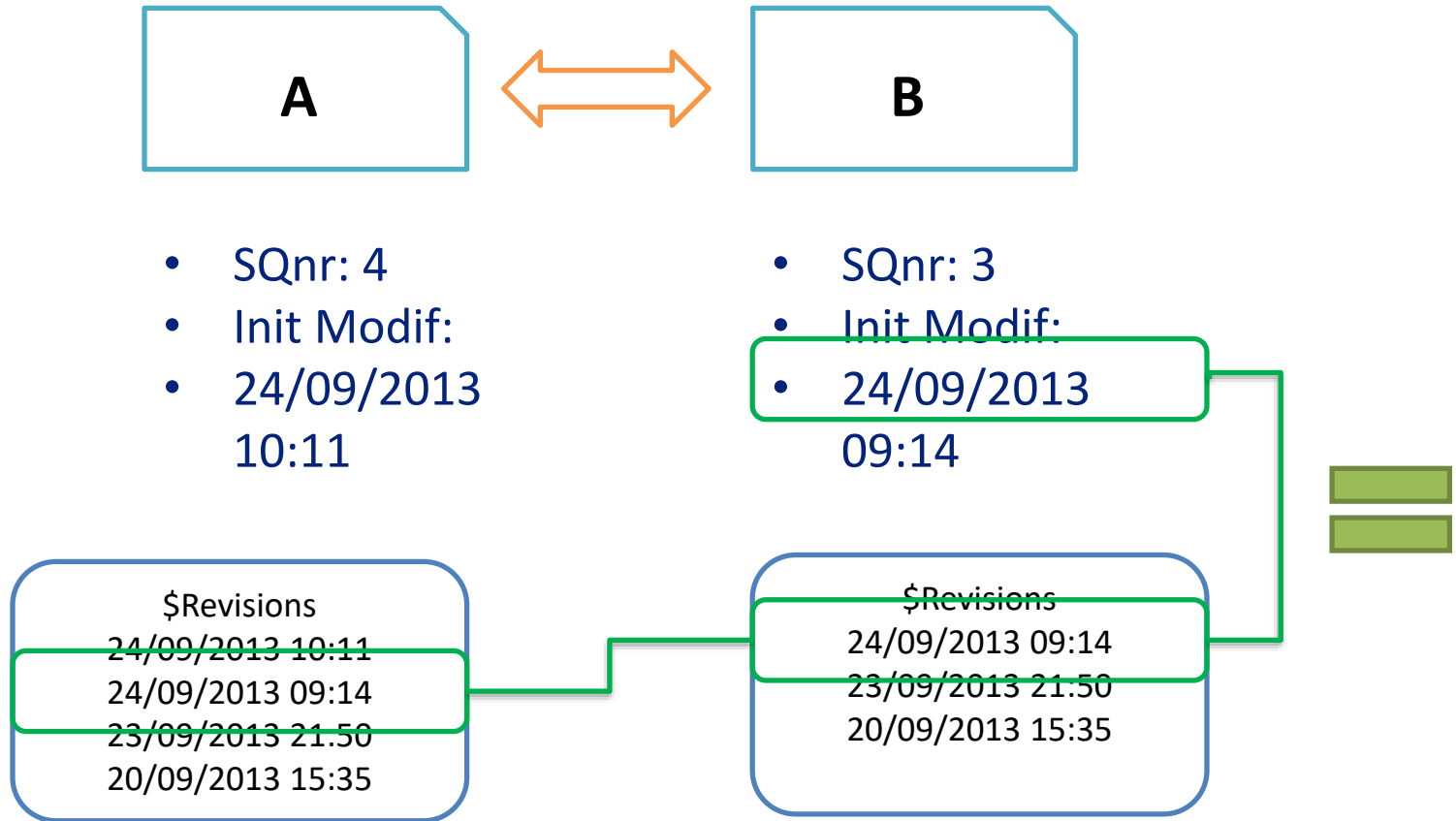
Replication – How comparison happens

2/3.) A and B are both Documents or Profiles (and they both exist)



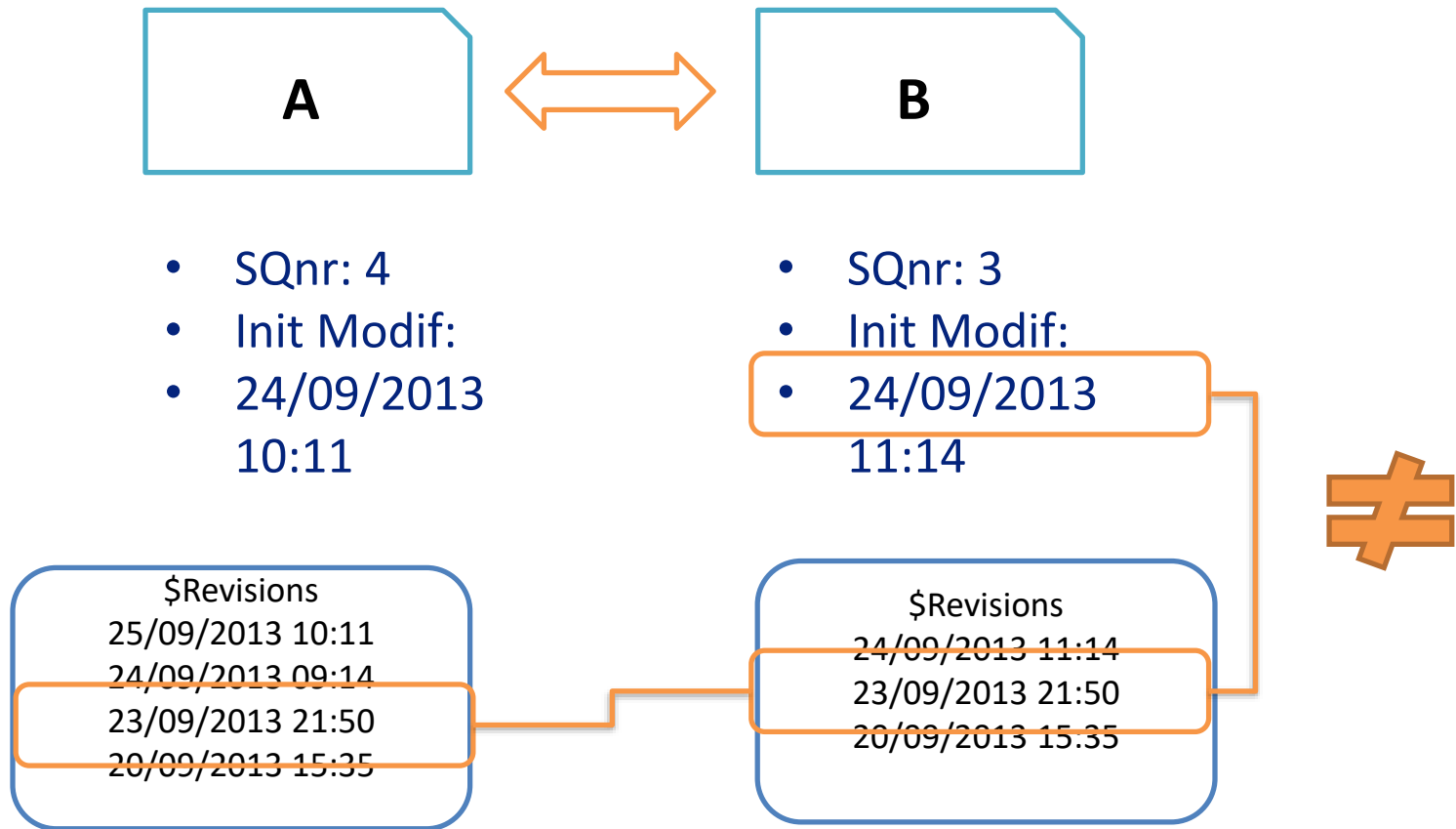
Replication – How comparison happens

- Scenario 1: Simple update
- Result: B is updated



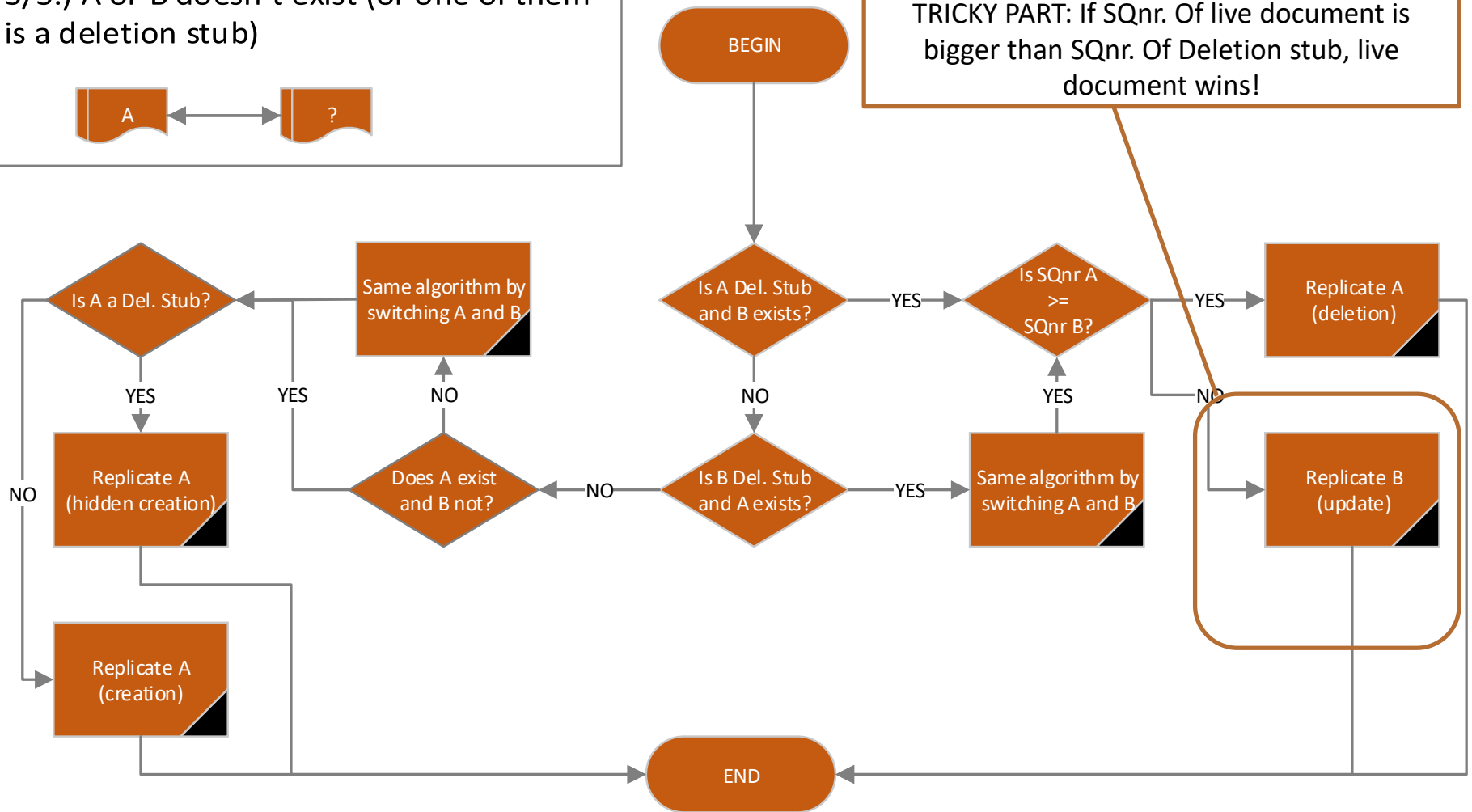
Replication – How comparison happens

- Scenario 2: Conflict
- Result: A wins conflict



Replication – How comparison happens

3/3.) A or B doesn't exist (or one of them is a deletion stub)

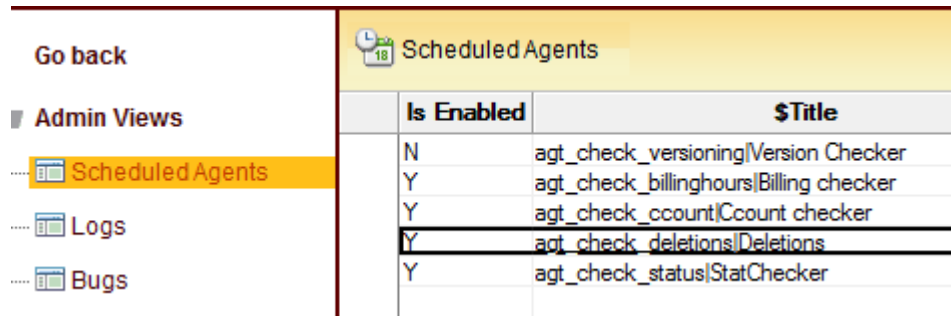


Beyond Documents: Views & Folders

- Views & Folders are both View Class notes
- Using the \$FormulaClass item in views
 - Decides what the view will display (default value = 1: documents only)
 - Change this value to the value found here:
<http://www.nsftools.com/tips/NotesTips.htm#formulaclass>
 - Create a view that displays (scheduled) agents, forms, views and more.
- An example:
 - Use 512 in \$FormulaClass to display filter class notes (agents)
 - Use the Selection Formula: `SELECT @Contains($Flags;"f") & @Contains($Flags;"S")`
 - To display Scheduled Agents
 - Column to Show if Agents are enabled:
 - `@If(@Contains($AssistFlags;"E");"Y";"N")`
 - Column to display name + alias
 - `@If(@Contains($TITLE;"|");@Implode(@Trim(@Explode($TITLE;"|"));"|");@Trim($TITLE)+"|" + @Trim($TITLE))`

Beyond Documents: Views & Folders

- \$Formulaclass (Continued)
 - Use formula \$OnBehalfOf to display well – the run on behalf of information.
- What you get is:

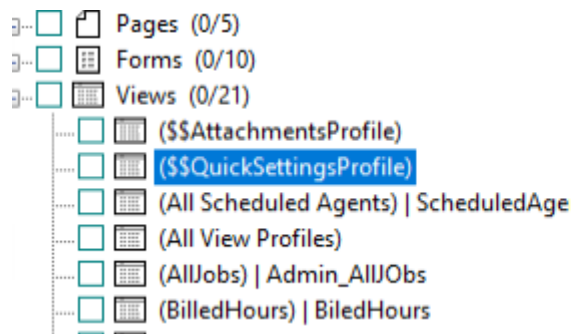


	Is Enabled	\$Title
	N	agt_check_versioning Version Checker
	Y	agt_check_billinghours Billing checker
	Y	agt_check_ccount Ccount checker
	Y	agt_check_deletions Deletions
	Y	agt_check_status StatChecker

- Bonus: with those agents in views, you can even program buttons to run them on server whenever you feel like it!

Beyond Documents: Views & Folders

- Using views as profile documents
 - Using profile documents can be troublesome due to caching on both the server and the client side
 - Have you ever tried using views as profile documents?
 - Views are the ONLY designs who's UNID you can get using their name:
 - `profileUNID = db.Getview(CONST_PROFILEVIEWNAME).UniversalID`
- Steps to set up
 - 1.) Create view with an easily recognizable name
 - 2.) Use scanEZ or LS to create profile items



Name	N.	S..	D.	I...	Value
\$UpdatedBy	1	4	0	3	CN=John Admin/O=OSCORP
\$ViewFormat	1	4	0	7	[n/a]
ab p_nabpath	1	4	0	12	names.nsf
ab p_salariesdbpath	1	4	0	13	app\salaries.nsf
ab p_salariesdbserver	1	4	0	14	Oscorp/OSCORP
ab p_ServerInfo	1	4	0	11	Oscorp/OSCORP

Beyond Documents: Views & Folders

- Steps to set up (Continued)
 - 3.) Use an implementation like this:

```
'View Profile constants
Const PROF_QUICKSETTINGS = "($$QuickSettingsProfile)"
'Quick Settings items
Const PROF_QUICKSETTINGS_SALARIESDBPATH = "p_salariesdbpath"
Const PROF_QUICKSETTINGS_SALARIESDBSERVER = "p_salariesdbserver"
Public Sub btn_getsalariesdb
'// Declarations
Dim s As NotesSession
Dim db As NotesDatabase
Dim profile As NotesDocument
Dim dbpath As String, dbserver As String
'// Initialize
Set s = New notessession
Set db = s.Currentdatabase
'// Grab quick settings profile doc
Set profile = db.Getdocumentbyunid(db.Getview(PROF_QUICKSETTINGS).Universalid)
dbpath = profile.Getfirstitem(PROF_QUICKSETTINGS_SALARIESDBPATH).Values(0)
dbserver = profile.Getfirstitem(PROF_QUICKSETTINGS_SALARIESDBSERVER).Values(0)
MsgBox "Salaries DB Path = "+dbserver+"!!"+dbpath
End Sub
```

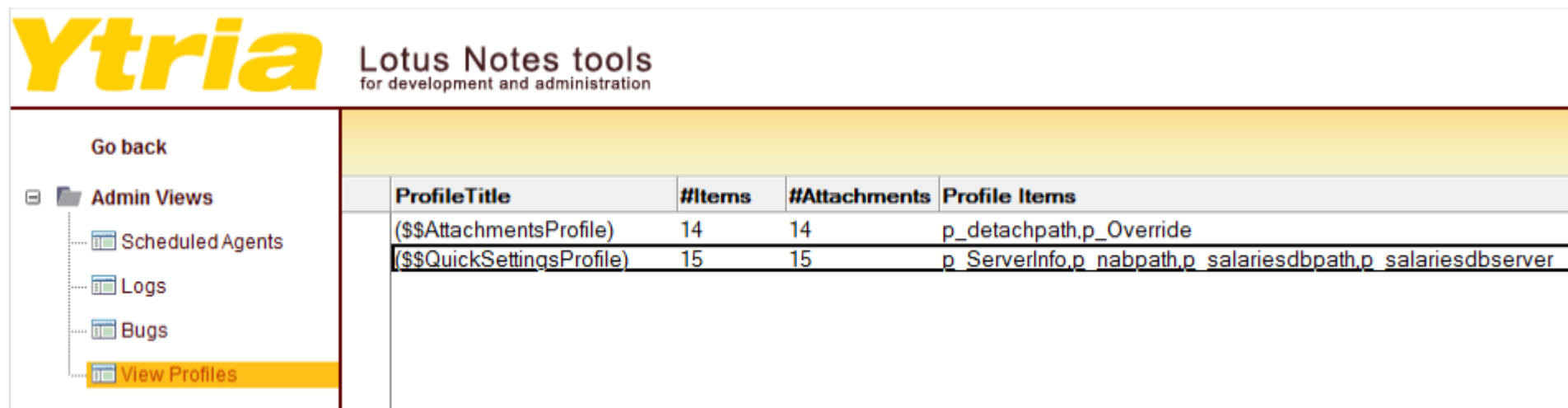
Beyond Documents: Views & Folders

- Bonus: you can even set up a view using a modified `$Formula` class item that lists all 'profile' views:
 - `$Formula` class item value for views: 8
 - Selection formula to only recognize views with e.g. “(\$\$ ”... In their names
 - Column using formula `$Title` to display profile title
 - Column using formula `@Elements(@DocFields)` to display # of items
 - Column using `@Attachments` to show # of attachments
 - Column using the following formula to display all profile items:

```
@Trim(@Transform(@Docfields;"var";  
@If(@Begins(var;"p_");  
var;  
""))));
```
- What you get is a list of profile views, with all profile items so you have an overview of all settings.

Beyond Documents: Views & Folders

- List of profile views:

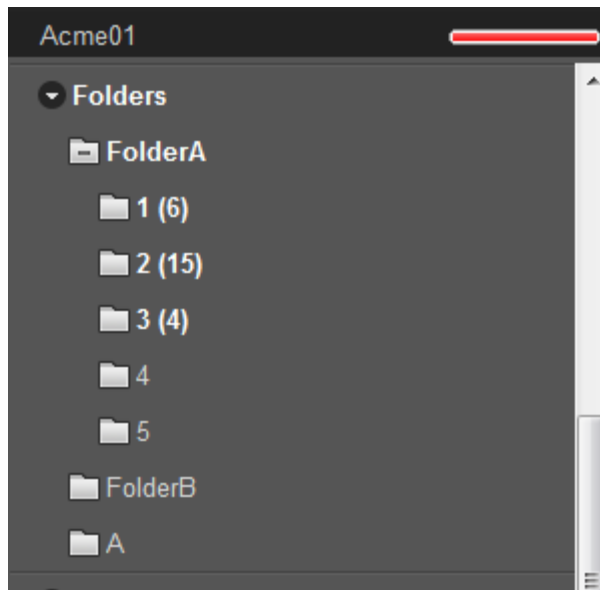


The screenshot shows the Ytria Lotus Notes tools interface. On the left, there is a navigation pane with a 'Go back' button and a folder icon for 'Admin Views'. Under 'Admin Views', there are four items: 'Scheduled Agents', 'Logs', 'Bugs', and 'View Profiles'. The 'View Profiles' item is highlighted in yellow. The main content area displays a table with the following data:

ProfileTitle	#Items	#Attachments	Profile Items
(\$AttachmentsProfile)	14	14	p_detachpath,p_Override
(\$QuickSettingsProfile)	15	15	p_ServerInfo,p_nabpath,p_salariesdbpath,p_salariesdbserver

Beyond Documents: Views & Folders

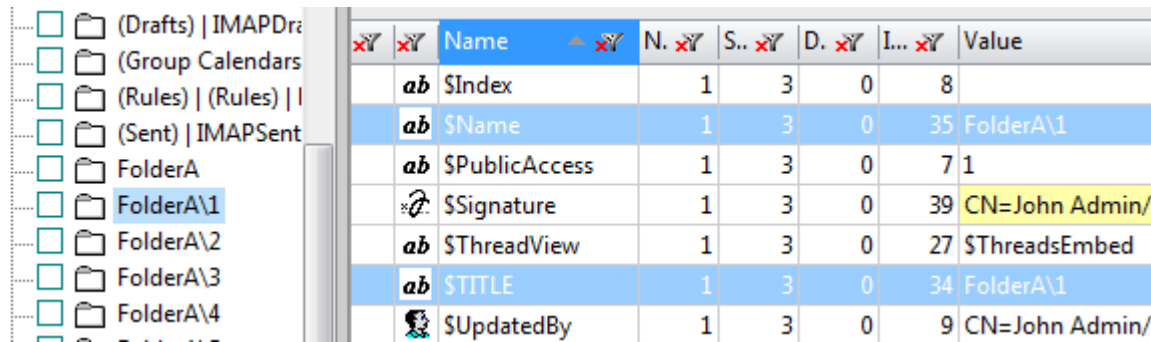
- About folders: customer scenario
 - About 1000 folders in a folder structure like this:



- User wants to move folderA\1 ... folderA\1000 to a new structure, FolderB
- Can't do it manually

Beyond Documents: Views & Folders

- About folder structures:
 - Turns out it's their NAME that's responsible for their structure:

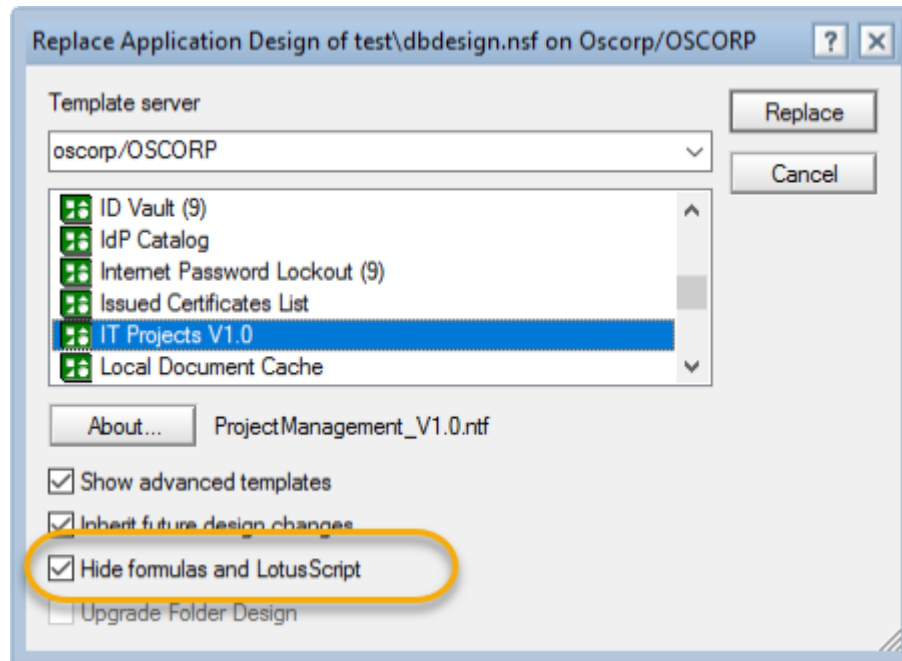


Name	N.	S.	D.	I...	Value
\$Index	1	3	0	8	
\$Name	1	3	0	35	FolderA\1
\$PublicAccess	1	3	0	7	1
\$Signature	1	3	0	39	CN=John Admin/
\$ThreadView	1	3	0	27	\$ThreadsEmbed
\$TITLE	1	3	0	34	FolderA\1
\$UpdatedBy	1	3	0	9	CN=John Admin/

- How to automate this?
 - Create a view that contains folders (using \$FormulaClass="8")
 - Use a Selection formula to only display folder you'd like to move
 - Write a formula agent to do something like this
 - @SetField(\$Title;@ReplaceSubstring(\$Title;"FolderA";"FolderB"))
 - But what about documents?
 - Again: it's the folders that know who belongs to them.

Beyond Documents: Hidden Designs

- What happens when design is hidden?
 - 1.) Database flag 'Design Hidden' enabled. Can't open the DB in Designer any more
 - 2.) Decompiled LS source code is removed from the database
 - 3.) Formulas are encrypted.

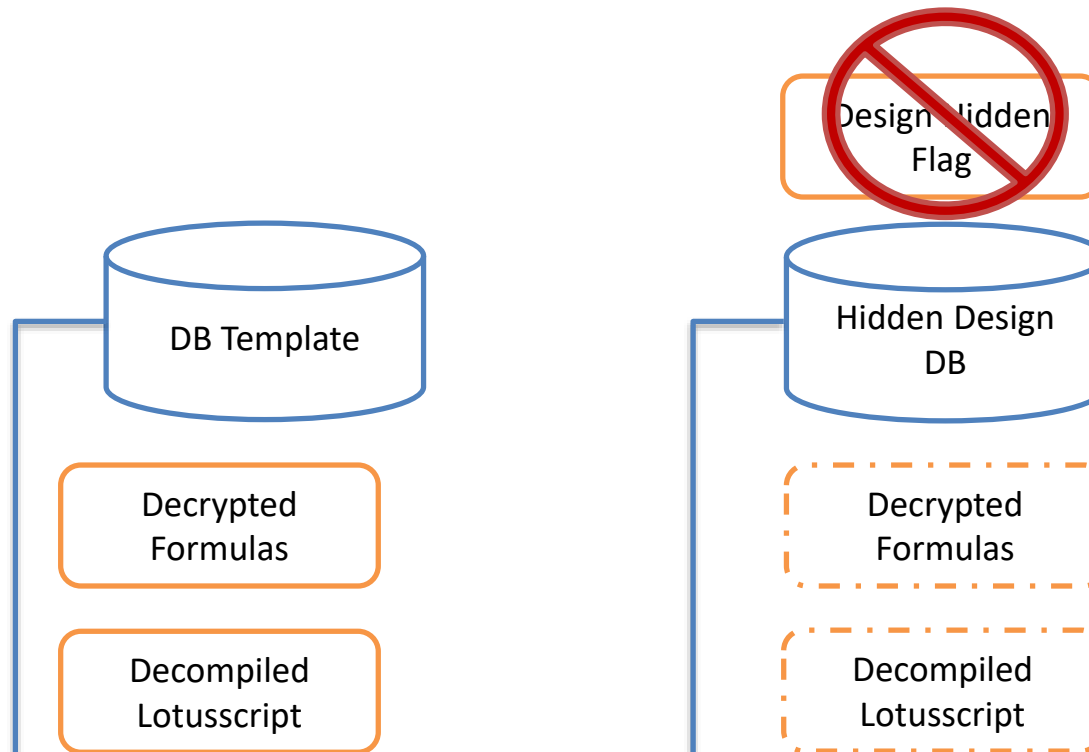


Beyond Documents: Hidden Designs

- Customer example
 - A developer created a database on server using a hidden template! Template is encrypted by their ID, they're no longer there and I can't access the DB design.
- Turns out: you can remove the DB design hidden flag!
 - Use the &H0020 for the hidden design flag for the NSFDbReplicaInfoGet API Call as described here (Eknori ;)):
 - <http://www.eknori.de/2008-10-23/sntt-is-database-design-hidden-notes-api-solution/>
 - Now you can open the DB in Designer!
- Sometimes, you'll get the entire DB design back, including LS and Formulas.

Beyond Documents: Hidden Designs

- As for the hidden code
 - If the developer chose to 'Refresh' the DB design instead of replacing it:



Secure Development

- Domino is a very secure platform – is it really?
- Local Domino Security is close to 0
 - Local protection is only enforced by the Notes client
 - Accessing local notes apps through the C API, access = Full Admin Access
 - Allows access to Reader & Author documents
 - Only security in place is encryption
 - Role of the 'Enforce Consistent ACLs' setting
 - If not enforced, no roles locally
 - And no local ACL
 - Do not allow client installation on server!

Secure Development

- Domino is a very secure platform – is it really?
- A few tips regarding secure development
 - Hiding Documents by giving no access to views is enough security
 - Even if no access to create views, users can create views locally.
 - Make sure to use readers & author fields and maintain a dynamic reader author list!
 - Funny story: Personal views
 - Make sure the server can ‘see them’ 😊
 - People with no ‘Replicate or Copy documents’ Flag
 - Can still use DXL export and import
 - Editor with ‘Write Public Access Documents’ but no Deletion rights
 - Let’s see!
 - What happened?
 - Can edit document, can therefore create the \$PublicAccess item with text value 1 – can delete document

Secure Development

- Conclusion: Security starts at Author
 - Maintain editions through agents users only invoke – do not let them edit documents directly
- Last funny (horror) story
 - Developer at a large Financial Institution
 - Bad relationship with Administrators
 - Has to ask for signatures on agents for his entire team
 - Finishes agent that recalculates documents
 - Puts innocent sounding subroutine in agent
 - Agent creates a new notes Doc in a DB without form
 - Creates a RT item
 - Uses the `ritem.Embedobject(C:\Program Files\IBM\Domino\Data\server.id)`
 - Get it yet? 😊