OPENNTF WEBINAR

September 18, 2025

Domino for Newbies: Development II

Jesse Gallagher, Graham Acres

AGENDA

- Welcome
- Presentation
  - Jesse Gallagher, Graham Acres
- Q and A

OPENNTF SPONSORS

- Prominic donates all IT related services
  - Cloud Hosting for OpenNTF
  - Infrastructure management for HCL Domino Servers

COMMUNITY – WE ARE ALL OPENNTF

- Participate – we all learn from each other
- We are all volunteers
- No effort is too small
- If your idea is bigger than you can do on your own, we can connect you to a team to work on it
- Test or help or modify an existing project
- Write guides or documentation
- Add reviews on projects / stars on Snippets

2025-2026 Elections Reminder

UPCOMING EVENTS

- October
  - OpenNTF Webinar: Creating a Service Oriented Architecture Platform with Domino – Richard Moy
    - October 16
  - Let's Connect - Nordics User Group
    - October 21, Copenhagen
    - October 23, Stockholm
    - https://letsconnect.world
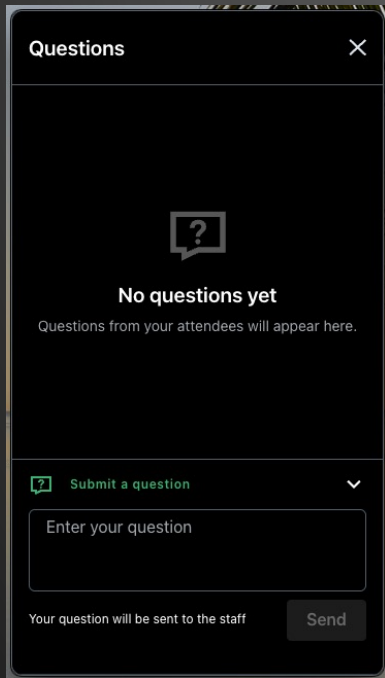 - Open Mic with the HCL Developers
    - TBD

JOIN US ON DISCORD

https://openntf.org/discord

ASKING QUESTIONS

- First Question – Will this be recorded?
  - Yes, view on YouTube!!!
  - https://www.youtube.com/user/OpenNTF
- Use the Questions Pane in GoToWebinar
- We will get to your questions at the end of the webinar
- The speakers will respond to your ques~~verbally~~
  - (not in the Questions pane)
- Please keep all questions related to topics that our speakers are discussing
- Unrelated Question => post at:
  - https://openntf.org/discord

Questions ✕

❓

No questions yet
Questions from your attendees will appear here.

Submit a question ⌄

Enter your question

Your question will be sent to the staff   Send

Share your Dev Tips!

PRESENTATION

Domino for Newbies: Development II

YOUR PRESENTERS

Jesse Gallagher

- Domino dev since around V7
- HCL Lifetime Ambassador and former IBM Champion
- OpenNTF Board and IP Manager 2014-2025
- Philadelphia, PA, USA
- https://frostillic.us
- @jesse@pub.frostillic.us

Graham Acres

- Notes / Domino app architect since 1992
- HCL Ambassador, former IBM Champion
- Canadian User Group (C3UG)
- OpenNTF Board 2018-2025
- Vancouver, Canada

TOPICS

- Who is this presentation for?
- Views, Views, Views
- Roles
- Templates
- Toolbars
- Subforms
- Agents
- How Do I Find Stuff?
- Q&A

THIS PRESENTATION IS FOR…

- People who are new to the Domino world
  - You have a new job as a Domino developer
  - You inherit the dusty old server in the corner … and something just broke
- Some level of development experience
  - Excel Macros
  - Full Stack devs
  - Java devs

G

VIEW BASICS

Views in Domino

* One of the two primary client UI elements, along with forms

* Also used for web UIs in both legacy and XPages

* *Also* used for data access by essentially everything in Domino

	Web Path	NSF Path	Servers
✔	/blog	jesse/jblog.nsf	CN=Kalgan/O=Frost
✔	/browser	dev/jakartabrowser.nsf	CN=Kalgan/O=Frost
✔	/browser2	dev/jakartabrowser2.nsf	CN=Kalgan/O=Frost
✔	/fakemodule	i/do/not/exist.nsf	CN=FakeServer/O=Frost
✔	/magpie	jesse/gog.nsf	CN=Kalgan/O=Frost
✔	/maven	dev/mvn.nsf	CN=Kalgan/O=Frost
✔	/moduleapp	dev/jakarta/mod.nsf	CN=Kalgan/O=Frost
✔	/moduleapp2	dev/jakarta/mod2.nsf	CN=Kalgan/O=Frost
✘	/moduleexample	dev/jakarta.nsf	CN=Kalgan/O=Frost
✔	/opensteme	opensstf/mainiss.nsf	CN=Kalgan/O=Frost

What Is A View?

* Views have two primary components:

    * The selection formula, defining what documents can appear in the view

    * The columns, defining the data pulled from those documents and how it's presented

* They're *kind of* like SQL views, but differ greatly in mechanics

* Also similar to NoSQL map/reduce operations

Select Formulas

* Written using the Notes formula language, which is also used for commands and value computation

* Examples:

    * `SELECT @All` - selects all documents in the DB

    * `SELECT Form="Person"` - selects all documents using the "Person" form

    * `SELECT Form="Person" & Department="IT"` - same as above, but only where the "Department" item is "IT"

    * `SELECT Form="Person" | @AllChildren` - all "Person" documents, plus any immediate children regardless of form

* You can also use "Simple Search", which is a semi-graphical way to filter documents

    * It's best to get used to formulas, though

Select Formulas, cont'd

* It's a good idea to start all selections with Form="Whatever" unless you have a reason to do otherwise

    * Good for clarity, reduces unexpected inclusions, and is useful for DQL

* Never* use "current time" formulas: @Now, @Today, etc.

    * The view is rebuilt every time it's accessed, ruining performance

    * * you *can*, but only if you know what you're doing

* Also avoid user-specific formulas like @UserName

    * These will usually be the name of the server building the view

* You can't use @DbLookup and friends in view selections or columns

Columns

- Data from documents is extracted by defining columns

- In the basic case, this is one field from the document

- Can also be a formula, like `FirstName + " " + LastName`

- Can't* be a rich text or similar field

  - * usually
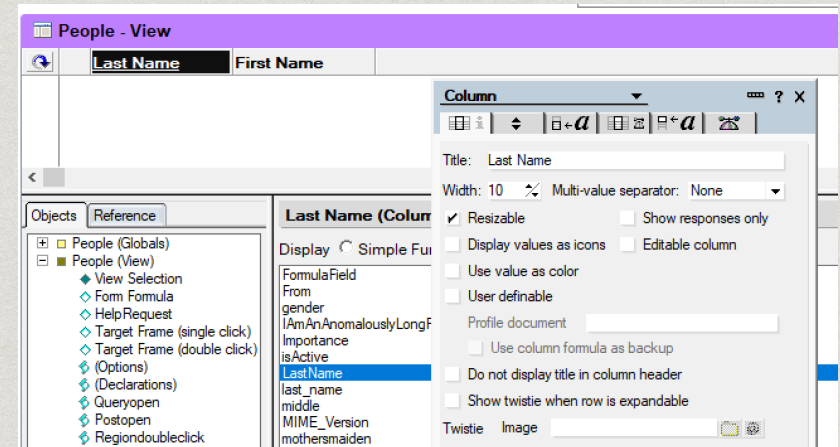
Columns, Side Tangent

* See the little "refresh" icon in the top left when editing a view in Designer?

* It's meant to show you a preview of the view contents inline, to help your work

* Do not trust it

    * Do not use it

        * It is a liar

            * And it always has been

Columns, cont'd

* Columns can be marked as sorted and, potentially, categorized

    * More on categories later

* Having sorted columns is needed for programmatic lookups by key

* "Totals" allows for some basic operations on number columns - total, average, etc.
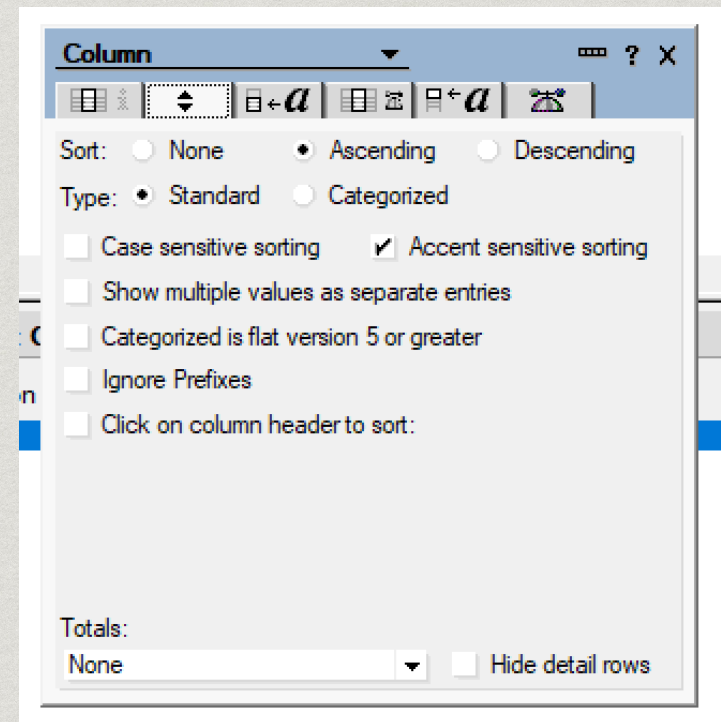
Columns, cont'd

* Categories significantly change the layout of the view

* They count as sorted keys for programmatic use, but also special category-specific needs

* For UI views, check "show twistie" for the column - everyone likes twisties

* Categories can have multiple levels

    * Values with a "\" in them are automatically split

    * Multiple columns can be categorized

	Last Name	First Name
▼HR		
	Bobson	Bob
▼IT		
	Fooson	Foo
	Steveson	Steve
▼Legal		
	Danson	Dan

Columns, cont'd

* Columns can be configured to show as an icon

  * A number represents a stock icon from the "Displaying an icon in a column" help document

  * A string represents an image resource in the view's DB
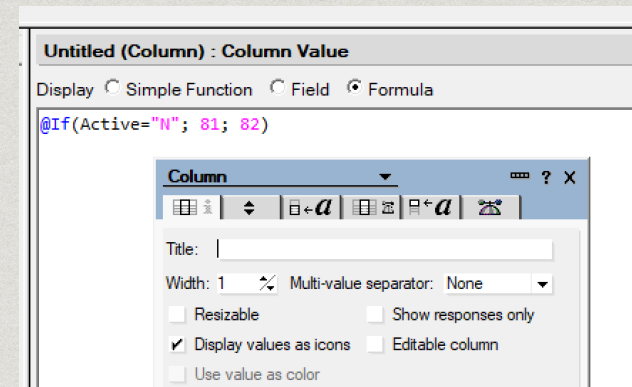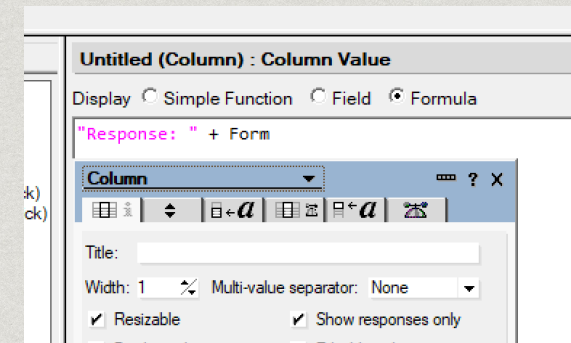
Columns, cont'd

- Columns can be configured to define the text and background color for subsequent columns

  - Best to mark this column as hidden

- Colors are red:green:blue number sets

  - Three numbers alone means just the text color

  - Six colors means background followed by text color

  - "-1:-1:-1" is "pass through"

- This is fun, but don't* go too nuts with it

  - * unless you're actually good at it
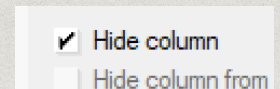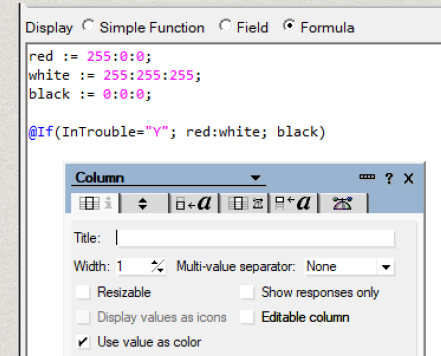
Display ○ Simple Function ○ Field ⦿ Formula

```
red := 255:0:0;
white := 255:255:255;
black := 0:0:0;

@If(InTrouble="Y"; red:white; black)
```

Column ▼ ⚊ ? X

Title: |

Width: 1   Multi-value separator: None ▼
☐ Resizable   ☐ Show responses only
☐ Display values as icons   ☐ Editable column
☑ Use value as color

☑ Hide column
☐ Hide column from

	Last Name	First Name
▼ HR		
✔	Bobson	Bob
▼ IT		
✔	Fooson	Foo
✔	Steveson	Steve
▼ Legal		
✖	Danson	Dan

Columns, cont'd

- Columns can be configured to apply specifically to response documents

  - You get one per view

  - Previous columns also apply to responses, but not subsequent ones

- These get an automatic indent, and can show a multi-level hierarchy

- They can be fiddly to get right

Folders

* Folders are very similar to views, but don't have a selection formula

* Documents are added to them either manually (like in the mail application) or programmatically (such as via agents)

* All other aspects of views apply: columns, categories, indexes, and so forth

VIEW INTERMEDIATES

View Indexes

- View indexes are generally maintained by the server

- In the normal case, you don't have to worry about it - make your view and it'll be kept up to date

- Some things can slow or trip up this process

  - Many documents

  - Too many categories

  - Very complex formulas

- Manually configure the index in special cases

- Use Shift-F9 to force a rebuild, or Ctrl-Shift-F9 to rebuild all views in the DB (takes a while, since it happens locally)
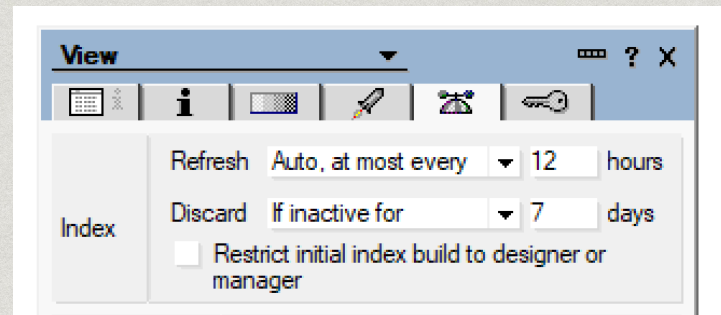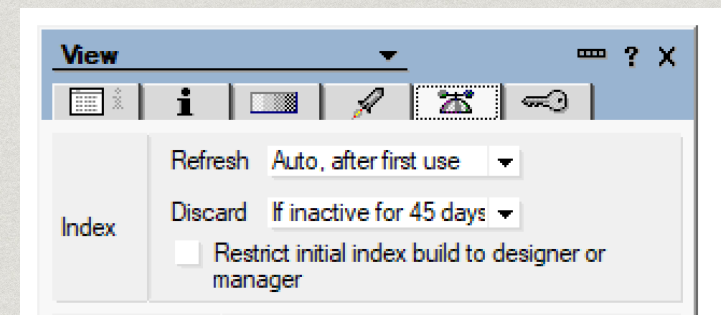
Response Documents

* Response documents have to be selected in the view

  * @AllChildren and @AllDescendants will specially select children of otherwise-selected docs

* Response documents without their parent in the view will not appear by default

  * Deselect "Show response documents in a hierarchy" to treat them the same as other docs

* In the UI, the selected doc is the target parent for new responses
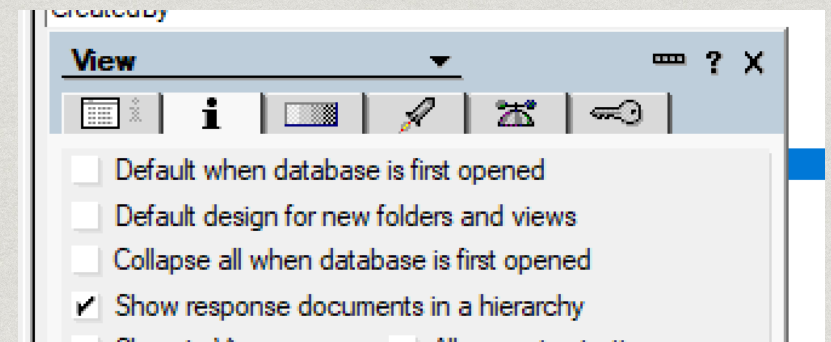
Utility Views

- It's usually a good idea to have some developer-focused "utility" views:

    - All documents in the DB with basic info, potentially sortable by created and modified date

    - All documents categorized by reader and author fields

    - Views for otherwise-hidden "keyword" or configuration documents

   - A "Trash" view for soft-deleted documents (more later)

#	Form	Size	Created	Modified
1	Person	1K	09/17/2025 11:01:28 AM	09/17/2025 11:01:34 AM
2	Person	1K	09/17/2025 11:01:35 AM	09/17/2025 11:08:01 AM
3	Person	1K	09/17/2025 11:02:49 AM	09/17/2025 11:02:54 AM
4	Person	1K	09/17/2025 11:03:01 AM	09/17/2025 11:03:21 AM
4.1	Comment	1K	09/17/2025 11:31:30 AM	09/17/2025 11:31:31 AM
4.1.1	Comment	1K	09/17/2025 11:31:36 AM	09/17/2025 11:31:37 AM
4.2	Comment	1K	09/17/2025 11:31:58 AM	09/17/2025 11:31:59 AM

Hidden Views

* Naming a view with parentheses around the name (e.g. "(My Secret View)") will hide it from the default list

* These views can still be used programmatically or as embedded views, making this naming style perfect for that

* You (or any user) can access these views by holding Ctrl-Shift when opening the DB or using "Go To..." in the View menu or workspace icon

   * Most users won't do this, but you should know it's possible

Shared Columns

* A column can be exported as shared (or a new one can be made manually)

* This is very useful for common logic: colors/icons for specific document types, repeated details used in several views

* As with many things, don't go too overboard
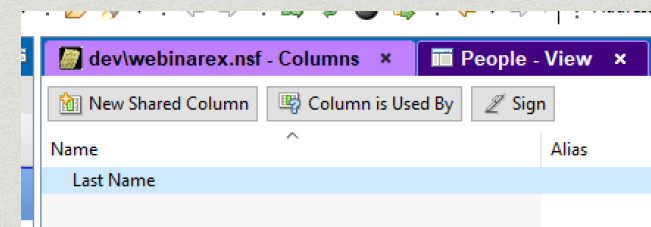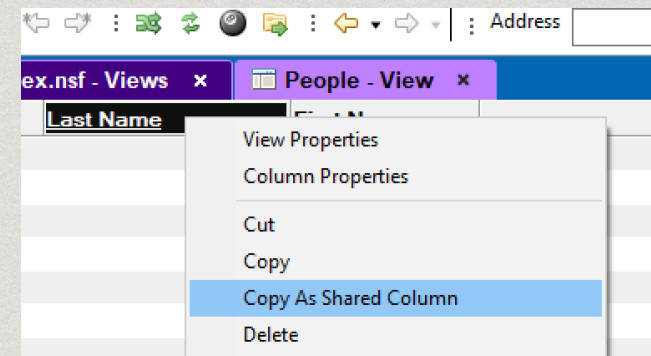
Embedded Views

- Any view can be embedded on a Form or Page

- This can be useful on its own, but pairs best with categorized views

- Specify a "Show single category" formula based on the document contents

    - Here, it's department name, showing all employees in that department
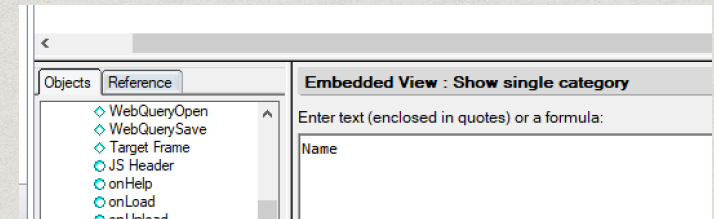
Calendar Views

- A view can be marked as displaying as a Calendar

- These are configured like normal views, but the first two columns are special:

  - The first contains the start date/time of the event

  - The second contains the event duration in minutes

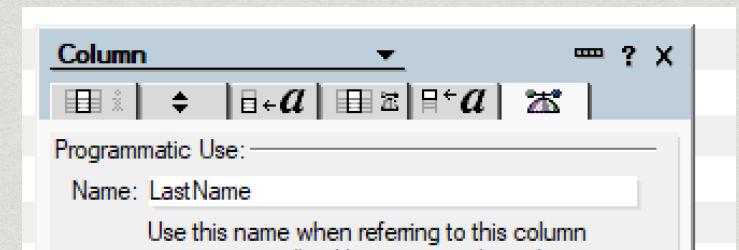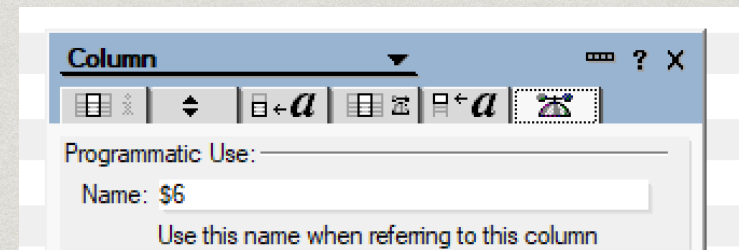- These mostly make sense for scheduling apps

Style | Calendar

Date/Time (Column) : Column Value

Display ○ Simple Function ○ Field ● Formula

@TimeMerge(EventDate; EventTime)

Duration (Column) : Column Value

Display ○ Simple Function ○ Field ● Formula

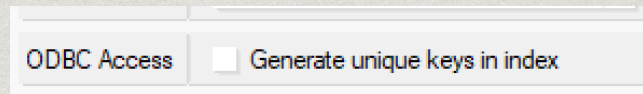DurationHours * 60 + DurationMinutes

VIEW ADVANCED

Programmatic Name

* View columns have a "programmatic name"

* This is usually automatically set by Designer, but you can technically manually set one

* This usually doesn't matter, but comes up in XPages and other programming environments

* You need to care about it in two cases:

  * Two columns can't have the same name - sometimes Designer messes up, duplicating the column contents

  * If the name matches an item on the document, the view will use the item value regardless of the column formula
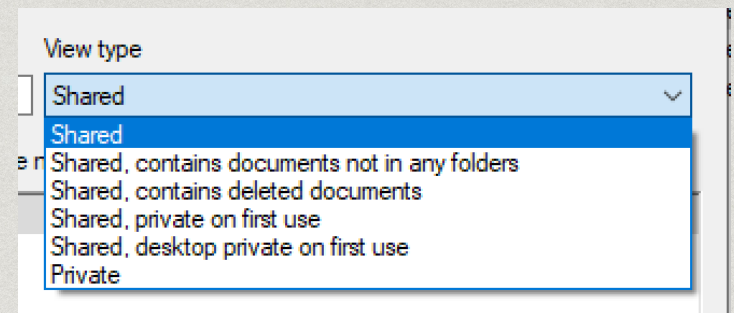

Column

Programmatic Use:
Name: $6
Use this name when referring to this column

Column

Programmatic Use:
Name: LastName
Use this name when referring to this column

Unique Keys

ODBC Access	☐ Generate unique keys in index

* This property is labeled "ODBC Access", but:

    * You don't need to know what "ODBC Access" means

    * It actually changes the way the view index is built significantly

* Views can have multiple entries per "Key" - that is, the combination of all sorted columns in sequence

* Setting this will (pretty much) make it so that only the first row per key is in the view

* It's a bit wonky sometimes - either ignore it entirely or use it sparingly

Special View Types

* By default, views are "shared": they contain all matching docs and all users see the same view

* You can also select docs not in folders - mostly useful for the mil app

* You can also show soft-deleted documents (i.e. a "Trash" view)

* Private views are fiddly

  * "Private on first use" makes a copy on the server only the user sees

  * "Desktop private on first use" is like that but for only one Notes install

  * "Private" is for yourself only

  * You will likely rarely or never need these, and they come with caveats

View type

Shared	⌄
Shared	
Shared, contains documents not in any folders	
Shared, contains deleted documents	
Shared, private on first use	
Shared, desktop private on first use	
Private	

ROLES

- View/form access
- Forms: Hide-when and controlled-access sections
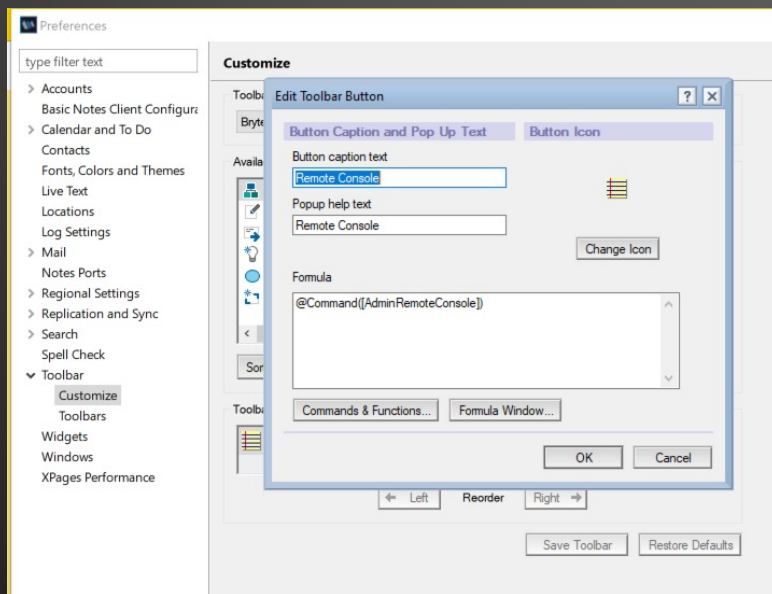- View column hide-when
- @UserRoles

TEMPLATES

- How does inheritance work?
- Create a new template from an existing database
- Create a new database from a template
- Using inheritance for version control

TOOLBARS

- Standard Toolbars
  - File > Preferences > Toolbars
- Customizing Toolbars

SUBFORMS

- Computed Subforms

AGENTS

- Understanding Agent Types
  - Formula
  - Simple Actions
  - LotusScript
  - Imported Java
  - Java

HOW DO I FIND STUFF?

- How to search the existing code across an NSF for a server name, not agent by agent. Like when migrating servers and someone hard coded an old server name.

- Errors in the log
  - Server Log is Console logging
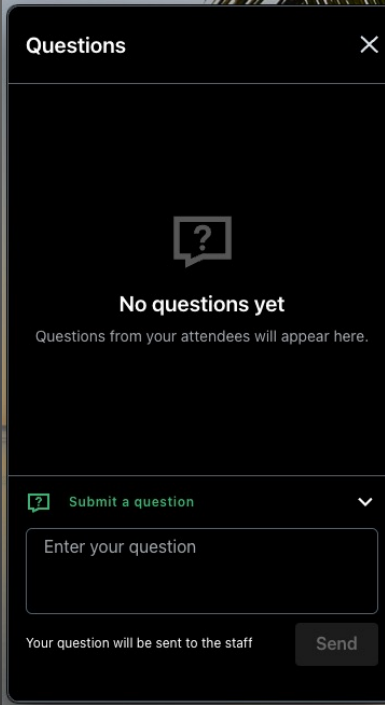   - Searching may be a challenge
  - Is there an Agent Log?
  - https://ab1osborne.blogspot.com/2025/09/colour-coding-lognsf.html

QUESTIONS?

Use the GoToWebinar Questions Pane

Please keep all questions related to the topics that our speakers are discussing!!!

Unrelated Question => post at:

https://openntf.org/discord