



HCL Notes Designer: struggling with configuration variables

PART 3 /4 : DOCUMENTS

Document vs Profile - Document

- ▶ Easy access
 - ▶ Access via views (a document can be selected in a view formula)
 - ▶ You can decide not to replicate these documents (Replication Formula)
 - ▶ The ACL can secure modification access
 - ▶ Multi-documents for configuration
 - ▶ System
 - ▶ Application Level
 - ▶ Document Level
 - ▶ Can be indexed or not (Full-text index)
 - ▶ Access in edit mode can be locked (cb properties)
- ▶ Issues
 - ▶ Need code to create the document (Roles, rights, etc.)
 - ▶ Performance can be impacted if accessed by views
 - ▶ UniqueKey needs coding (no duplicate for instance)
 - ▶ Deletion
 - ▶ Can be deleted by error or voluntary

Document vs Profile - Profile

- ▶ Easy access
 - ▶ Performance: you access directly the profile
 - ▶ Implicit security
 - ▶ No standard access by the user
 - ▶ UniqueKey is managed by Notes/Domino
- ▶ Pitfalls
 - ▶ Cannot list the profile in a view
 - ▶ Audit can be complicated
 - ▶ No full-text search
 - ▶ Difficult to manage versioning
 - ▶ Difficult to lock a profile

Document vs Profile - Choice

- ▶ Document

- ▶ The application needs multiple distinct configurations
- ▶ You need to audit/trace any change in configuration
- ▶ Important volume of data
- ▶ Performance is not critical

- ▶ Profile

- ▶ Configuration by user or context
- ▶ Frequent access and performance is critical
- ▶ You do not have to display the information

Documents as configurations docs: the basics

- ▶ Limitations
 - ▶ Notes field limits
 - ▶ It is a document, you can access a document with a view selection
 - ▶ A form is optional: only create one if you want the user to fill-in multiple fields
 - ▶ Recommended for debugging purpose!
 - ▶ The configuration is counted in number of documents in database
- ▶ Tools available to control documents:
 - ▶ HCL Notespeek
 - ▶ Ytria scanEZ
 - ▶ HCL Notes Panagenda « Advanced properties »
 - ▶ Beginning from R12 of HCL Notes

Documents as configuration docs

- ▶ Use another field in conjunction with Form field: ie: Type=Form
- ▶ Add an other item to tag the Document as a Configuration Doc:
 - ▶ DocType="Configuration"
 - ▶ Ignore these types of documents in the application views
- ▶ You can add \$PublicAccess="1" to make the configuration doc available to public access user (ACL option)
- ▶ Way To access the configuration doc
 - ▶ Views/Folders
 - ▶ Db.search
 - ▶ Formulas

Documents as configuration docs

- ▶ Formula you can use to get the configuration doc
 - ▶ @Dblookup
 - ▶ @DbColumn
 - ▶ @GetDocField (you will need the UNID)
- ▶ LotusScript/Java: Choose a method to access that document
 - ▶ NotesDatabase.Search
 - ▶ NotesView.GetDocumentByKey

Documents - Formula

- ▶ For server-side you can investigate DQL (starting with V11)
- ▶ @dblookup and @dbcolum
 - ▶ Need of a "Configuration" view or folder or anti-folder
- ▶ @getDocField
 - ▶ If you can get a Notesdocument unique ID, you can use this function to retrieve a field value

Documents - LotusScript

- ▶ Notesdatabase.Search
 - ▶ No impact if the database is full-text indexed or not
 - ▶ The query formula:
 - ▶ Formula= | |
- ▶ @dblookup and @dbcolum
- ▶ Need of a "Configuration" view or folder or anti-folder
- ▶ @getDocField
 - ▶ If you can get a Notesdocument unique ID, you can use this function to retrieve a field value

Documents – Special documents

- ▶ Special documents are internal documents not listed in documents count
 - ▶ Design elements
 - ▶ \$DbIcon: Notes database icon
 - ▶ \$Templatebuild: if exists contains versioning of the application
 - ▶ Views, etc.
- ▶ You can access 'special documents' via two methods
 - ▶ DesignCollection
 - ▶ A custom view
 - ▶ Create a "(\$Design)" view: viewname does not need to be \$Design
 - ▶ Create a column (sorted first column) on field \$Title
 - ▶ Select formula: select @isavailable(\$Title)
 - ▶ You can adapt if not interested in web-inf documents
 - ▶ `SELECT @IsAvailable($Title) & !(@Contains($Title;"WEB-INF") | @Contains($Title;"") | @Contains($Title;"Internal"))`
 - ▶ \$Flags="wY"
 - ▶ Use NotesDatabase.getView
 - ▶ Set doc=db.getView("(\$Design)")
 - ▶ Call doc.replaceItemValue("\$Flags","wY")
 - ▶ Call doc.save(true,false,false)