HCL Notes Designer: struggling with configuration variables

PART 2 /4 : PROFILES

Profiles: the basics

- Limitations
  - Notes field limits
  - Not really a document, you cannot access a profile 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!
- Tools available to control profiles:
  - HCL Admin Tools
  - Ytria scanEZ
  - HCL Notes Panagenda « Advanced properties »
    - Beginning from R12 of HCL Notes

Profiles: the basics cont'd

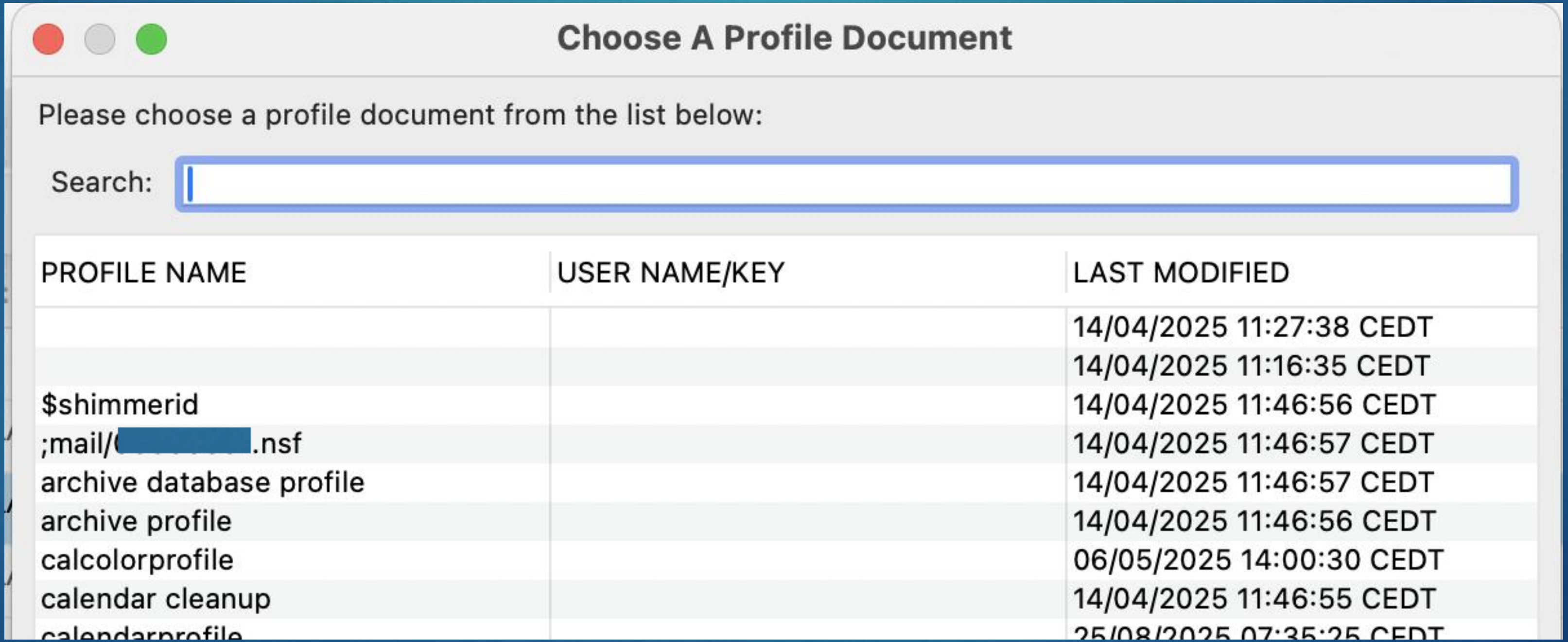
- Is created once you access the profile by code that « creates » a profile or a field in the profile
  - Can be empty if your code does not set any value
- Replication
  - Profiles replicate even if there is a replication formula selection that does not include them
  - No replication conflicts occur
- Used
  - Mainly to store global information and user defined informations
  - Ability to store specific user defined information
  - API or accounts key informations (BEWARE OF SECURITY)

Profiles: the basics cont'd

- Cache
  - Profiles are cached
    - To update the back-end data of a profile, the database needs to be closed and the client restarted
      - Formulas are concerned
      - Work-around exists for LS/Java code
    - Can be tricky on Web/xPages applications
      - Use an agent fo update the profile not a user UI form
    - On a server a profile can be cached within different threads
  - Do not use for dynamic and frequent updates
    - Avoid using it for increasing a sequenced number

Profiles: the basics cont'd

- Samples
  - The best HCL Notes Database sample is the mail template (mail12.ntf, mail14.ntf, etc…)



Choose A Profile Document

Please choose a profile document from the list below:

Search:

PROFILE NAME	USER NAME/KEY	LAST MODIFIED
		14/04/2025 11:27:38 CEDT
		14/04/2025 11:16:35 CEDT
$shimmerid		14/04/2025 11:46:56 CEDT
;mail/[redacted].nsf		14/04/2025 11:46:57 CEDT
archive database profile		14/04/2025 11:46:57 CEDT
archive profile		14/04/2025 11:46:56 CEDT
calcolorprofile		06/05/2025 14:00:30 CEDT
calendar cleanup		14/04/2025 11:46:55 CEDT
calendarprofile		25/08/2025 07:35:25 CEDT

Profiles – Formula Language

- Profilename
  - Generally a Form name
    - Even if the Form does not exist
    - Should be unique: avoid using an existing normal form (to avoid confusion)
      - You can prefix Profiles with ‘p_’ to indicate the form is a profile
        - Ie for a form name used in profiles: (p_parameters) | p_parameters
- The uniqueKey parameter can be whatever you want as TEXT
  - @username: specific to a user
  - Context: whatever value in a specific context
    - Country
    - Department
    - Etc.

Profiles – Formula Language cont'd

- Setter
  - @SetProfileField("ProfileName"; "Fieldname"; Value [; "uniqueKey"])
- Getter
  - Value := @GetProfileField("ProfileName"; "Fieldname" [; "uniqueKey"])
- Edition of a profile
  - Needs a Form for dispklay and input
  - **@Command([EditProfile]** ; "ProfileName" ; "uniqueKey" **)**
    - Is executed after ALL other @functions
  - **@Command([EditProfileDocument]** ; "ProfileName" ; "uniqueKey" **)**
    - Is executed immediately

Profiles - LotusScript

- Cache management
  - Using notesdocument.remove(true)
    - Before calling the creation of the profile
      - Needs backup of all items before
        - Not a good solution
  - Drilling the profile collection with **GetProfileDocCollection**
    - This way you really get the profile not the cached one
- Profile document is a NotesDocument
  - Save it after updates
  - All methods and properties of a NotesDocument are available

Profiles - LotusScript

- The profile document always exists with the getProfileDocument function
  - You need to test items values to see if a profile really exists
    - Set a field 'isvalid' to a value of "1" to be sure data was set in the profile
- Getters
  - A new empty profile is created it it does not exist
  - NotesDatabase.**getProfileDocument(**"ProfileName" ; "uniqueKey"**)**
    - Set profileDoc = db.getProfileDocument **(**"Parameters"**)**
      - If profileDoc.getItemValue("isValid")(0)="1" then…
        - isValid is an item you define
  - NotesDatabase.getProfileDocCollection(["ProfileName"])
    - Count property represents the number of profiles found
    - Without parameter you retrieve all profiles in a collection

Profiles - LotusScript

- Setter
  - Once you get a profile, just use the notesDocument.**Save** method
- Edit
  - You can use *notesUIWorkspace* **.EditDocument** with a profileDoc instance to open the profile

Profiles – LotusScript

- You can add the field $PublicAccess with a value of "1" to allow access to public users
- Class
  - You can create a class in a library to manipulate profiles:
    - New method find and load a profilename
    - Etc.

Profiles – Java

- Java mimics the LS functions
  - You do have to save the profile
- Setter/Getter
  - The function get an instance of a specific profile, it is created if needed
  - public Document getProfileDocument("ProfileName","uniqueKey")
- Getter
  - To scan all profiles
  - public DocumentCollection getProfileDocCollection(["ProfileName"])

Profiles – Cache workarounds

- Use the collection to retrieve the profile document instead of cached profile
- A proposed solution:
  - Put the profiles in a configuration database
    - The application gets the information from the configuration database
    - Not available for formula language
      - Formula language can only access the profiles from the current database
    - Solution for LS or Java