

Form Information

Name:

Last Modification:
 Comment:
 Type:
 Include in Compose Menu:
 Include in Query by Form:
 Default Database Form:
 Automatically Refresh Fields:
 Mail New Documents When Saving:
 Store Form In Documents:
 Inherit Default Field Values:
 Updates Become Responses:
 Retain Prior Versions As Responses:
 Activate Objects When Composing:
 Activate Objects When Editing:
 Activate Objects When Reading:
 Document Encryption Keys:
 Composed Documents May Be Read By:
 Form May Be Composed By:

dlgExternalQA

01/11/2023 02:55:15 PM
 [Not Assigned]
 Document
 No
 No
 No
 No
 No
 No
 No
 No
 No
 No
 No
 [None Assigned]
 All Users
 All Users

Subcomponents:

Field:

Datatype:
 Input Multi-Value Separator(s):
 Display Multi-Value Separator:
 Help Description:
 Field Type:
 Sign When Mailing/In Section:
 Encryption:
 Update Requires Editor Access:
 Formula:

DAList

Text
 Comma, Semicolon, New Line
 New Line
 [Not Assigned]
 Computed when composed
 No
 Disabled
 No
 val := @DbLookup("" :
 "NoCache"; "" ; "config"; "config";
 "DAList");ret :=
 @If(@IsError(val); "< Error
 Retrieving Server Names >";
 @Word(val; "~"; 1));
 @Trim(@If(DAListExternal = "";
 ret; @Replace(ret;
 DAListExternal; "")));
 No

Is Scripted:

Field:

Datatype:
 Input Multi-Value Separator(s):
 Display Multi-Value Separator:
 Help Description:
 Field Type:
 Keyword User Interface:
 Allowable Keywords:
 Sign When Mailing/In Section:
 Encryption:
 Update Requires Editor Access:

Identifier

Keywords
 Comma
 Comma
 [Not Assigned]
 Editable
 Check Boxes
 Additional Directory|d
 Org/OrgUnit in Notes Name|o
 No
 Disabled
 No

Is Scripted:	No
Field:	DAListExternal
Datatype:	Keywords
Help Description:	[Not Assigned]
Field Type:	Editable
Keyword User Interface:	Standard
Allow Values Not In List:	No
Allowable Keywords Formula:	values := DAList;list := @If(@IsError(values); ""; @ReplaceSubstring(@Replace Substring(values; " "; "!"); "~"; " "));list;
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Is Scripted:	No
Field:	OrgsExternal
Datatype:	Keywords
Input Multi-Value Separator(s):	Comma
Display Multi-Value Separator:	Comma
Help Description:	[Not Assigned]
Field Type:	Editable
Keyword User Interface:	Standard
Allow Values Not In List:	No
Allowable Keywords Formula:	val := @DbLookup("" : "NoCache"; "" ; "config"; "config"; "Orgs");@If(@IsError(val); "< Error Retrieving Orgs >"; val);
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Is Scripted:	No
Hide When:	@IsNotMember("o"; Identifier);
JavaScript & HTML Code:	
[None]	
LotusScript Code:	
[None]	
Name:	dlgLicenseInfo
Last Modification:	01/11/2023 02:55:15 PM
Comment:	[Not Assigned]
Type:	Document
Include in Compose Menu:	No
Include in Query by Form:	No
Default Database Form:	No
Automatically Refresh Fields:	No
Mail New Documents When Saving:	No
Store Form In Documents:	No
Inherit Default Field Values:	No
Updates Become Responses:	No
Retain Prior Versions As Responses:	No
Activate Objects When Composing:	No
Activate Objects When Editing:	No
Activate Objects When Reading:	No
Document Encryption Keys:	[None Assigned]
Composed Documents May Be Read By:	All Users

Form May Be Composed By:
Subcomponents:

Field:

Datatype:
Help Description:
Field Type:
Sign When Mailing/In Section:
Encryption:
Update Requires Editor Access:
Formula:
Is Scripted:

Field:

Datatype:
Help Description:
Field Type:
Sign When Mailing/In Section:
Encryption:
Update Requires Editor Access:
Formula:
Is Scripted:

Field:

Datatype:
Help Description:
Field Type:
Sign When Mailing/In Section:
Encryption:
Update Requires Editor Access:
Formula:
Is Scripted:

Field:

Datatype:
Help Description:
Field Type:
Sign When Mailing/In Section:
Encryption:
Update Requires Editor Access:
Formula:
Is Scripted:

Field:

Datatype:
Help Description:
Field Type:
Sign When Mailing/In Section:
Encryption:
Update Requires Editor Access:
Formula:
Is Scripted:

Field:

Datatype:
Number Format:
Percentage (value * 100)%:
Parentheses on Negative Numbers:
Punctuated at Thousands:
Help Description:
Field Type:
Sign When Mailing/In Section:

All Users

Name

Text
[Not Assigned]
Computed when composed
No
Disabled
No
@ThisValue
No

Email

Text
[Not Assigned]
Computed when composed
No
Disabled
No
@ThisValue
No

Phone

Text
[Not Assigned]
Computed when composed
No
Disabled
No
@ThisValue
No

CompanyName

Text
[Not Assigned]
Computed when composed
No
Disabled
No
@ThisValue
No

HCLAccount

Text
[Not Assigned]
Computed when composed
No
Disabled
No
@ThisValue
No

PersonDocCount

Number
General
No
No
No
[Not Assigned]
Computed when composed
No

Encryption:	Disabled
Update Requires Editor Access:	No
Formula:	@ThisValue
Is Scripted:	No
Field:	DAUserCount
Datatype:	Number
Number Format:	General
Percentage (value * 100)%:	No
Parentheses on Negative Numbers:	No
Punctuated at Thousands:	No
Help Description:	[Not Assigned]
Field Type:	Computed when composed
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Formula:	@ThisValue
Is Scripted:	No
Field:	ValidUserCount
Datatype:	Number
Number Format:	General
Percentage (value * 100)%:	No
Parentheses on Negative Numbers:	No
Punctuated at Thousands:	No
Help Description:	[Not Assigned]
Field Type:	Computed when composed
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Formula:	@ThisValue
Is Scripted:	No
Hotspot	
Type:	Action Hotspot
Caption:	Yes
Formula:	
	@Command([RefreshParentNot e]);
	@Command([CloseWindow]);
Hotspot	
Type:	Action Hotspot
Caption:	No
Formula:	@Command([CloseWindow]);
JavaScript & HTML Code:	
[None]	
LotusScript Code:	
[None]	
Name:	dlgUserInfo
Last Modification:	01/11/2023 02:55:15 PM
Comment:	[Not Assigned]
Type:	Document
Include in Compose Menu:	No
Include in Query by Form:	No
Default Database Form:	No
Automatically Refresh Fields:	No
Mail New Documents When Saving:	No
Store Form In Documents:	No

Inherit Default Field Values:	No
Updates Become Responses:	No
Retain Prior Versions As Responses:	No
Activate Objects When Composing:	No
Activate Objects When Editing:	No
Activate Objects When Reading:	No
Document Encryption Keys:	[None Assigned]
Composed Documents May Be Read By:	All Users
Form May Be Composed By:	All Users

Subcomponents:

Field:	Name
Datatype:	Text
Help Description:	[Not Assigned]
Field Type:	Editable
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Is Scripted:	No
Field:	Email
Datatype:	Text
Help Description:	[Not Assigned]
Field Type:	Editable
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Is Scripted:	No
Field:	Phone
Datatype:	Text
Help Description:	[Not Assigned]
Field Type:	Editable
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Is Scripted:	No
Field:	CompanyName
Datatype:	Text
Help Description:	[Not Assigned]
Field Type:	Editable
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Is Scripted:	No
Field:	HCLAccount
Datatype:	Text
Help Description:	[Not Assigned]
Field Type:	Editable
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Is Scripted:	No

JavaScript & HTML Code:

[None]

LotusScript Code:

[None]

Name: fmAgreement

Alias:	agreement
Last Modification:	01/11/2023 02:55:18 PM
Comment:	[Not Assigned]
Type:	Document
Include in Compose Menu:	No
Include in Query by Form:	No
Default Database Form:	No
Automatically Refresh Fields:	No
Mail New Documents When Saving:	No
Store Form In Documents:	No
Inherit Default Field Values:	No
Updates Become Responses:	No
Retain Prior Versions As Responses:	No
Activate Objects When Composing:	No
Activate Objects When Editing:	No
Activate Objects When Reading:	No
Document Encryption Keys:	[None Assigned]
Composed Documents May Be Read By:	All Users
Form May Be Composed By:	All Users
Subcomponents:	
Layout Region:	
Show Border:	No
Three Dimensional Style:	No
Show Grid:	No
Snap to Grid:	Yes
Grid Size:	72
Field:	Untitled
Datatype:	Text
Help Description:	[Not Assigned]
Field Type:	Editable
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Is Scripted:	No
Field:	Accept
Datatype:	Keywords
Help Description:	[Not Assigned]
Field Type:	Editable
Keyword User Interface:	Radio Buttons
Allowable Keywords:	I understand and accept the above License Agreement Terms and Conditions 1 I understand and DO NOT accept the above License Agreement Terms and Conditions 2
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Default Value Formula:	"2"
Is Scripted:	No
Hotspot	
Type:	Button
Caption:	Continue
Is Scripted:	Yes, see below
Hide When:	Accept != "1";

Hotspot

Type:
Caption:
Formula:

Button
Cancel

Field:

Datatype:
Help Description:
Field Type:
Sign When Mailing/In Section:
Encryption:
Update Requires Editor Access:
Default Value Formula:

@PostedCommand([FileClose
Window]);

License

Text
[Not Assigned]
Editable
No
Disabled
No

nl := @NewLine;"GRANT OF
LICENSE:" + nl + nl + "1.
General Grant." + nl +
""Licensed Software" includes
each software product listed on
the first page of this Agreement
(a "Licensed Software Product")
and related documentation
including any updates, changes,
enhancements or modifications
thereto. Subject to the terms
and conditions of this
Agreement, for each Licensed
Software Product HCL
TECHNOLOGIES hereby grants
to Customer a personal,
non-exclusive, non-transferable
license to use the Licensed
Software Product for its own
internal business purposes for
the duration of the term set out
above (License Term) for such
Licensed Software Product. For
each Licensed Software
Product, Customer agrees to
pay to HCL TECHNOLOGIES or
its authorized reseller the
License Fee set out above for
such Licensed Software
Product." + nl + nl + "2. Use and
Restrictions." + nl + "If Source
Code Version is designated
above for a Licensed Software
Product then Customer is
entitled to the source code for,
and to modify such Licensed
Software Product, otherwise
Customer is not entitled to the
source code. Customer shall
not, and shall not permit others
to, reverse engineer,
de-compile, disassemble,
translate or modify such

Licensed Software Product. Customer shall be restricted to running the server component of each Licensed Software Product on the number of servers designated above (Number of Server Licenses) for such Licensed Software Product and no more. Customer shall be restricted to running the client component of each Licensed Software Product on the number of machines designated above (Number of Licensed Users) for such Licensed Software Product and no more. Customer shall be restricted to using each Licensed Software Product in connection with the number of databases designated above (Number of Database Licenses) for such Licensed Software Product and no more." + nl + nl + "3. Sublicense or Transfer." + nl + "Customer shall not use the Licensed Software as part of a service bureau arrangement or for the direct benefit of its customers or any other third party. Customer shall not sublicense, rent, transfer, or assign the Licensed Software or this Agreement without the prior written consent of HCL TECHNOLOGIES" + nl + nl + "4. Source Code." + nl + "If Source Code Version is designated above for a Licensed Software Product, Customer shall hold such source code in trust and confidence for and on behalf of HCL TECHNOLOGIES and shall not, except as expressly authorized hereunder or in writing by HCL TECHNOLOGIES, use, copy or disclose to any third party any part of the source code so received. Customer shall take appropriate action by instruction, agreement or otherwise to ensure that its directors, officers, employees, consultants and agents are required to keep confidential source code which is disclosed

to or comes into the possession of any of them." + nl + nl + "5. Modifications." + nl + "Unless expressly stated otherwise herein, all of the terms of this Agreement apply to any modified version of the Licensed Software including modifications made by HCL TECHNOLOGIES pursuant to a Consulting Proposal or by its authorized reseller." + nl + nl + "6. Other Software Licenses Required." + nl + "Customer acknowledges that the Licensed Software is designed to be used with other software programs and Customer agrees that it is solely responsible for obtaining and complying with all licenses to use such other programs from the relevant owners or licensors thereof. Customer shall be fully liable for any claims arising as a result of its failure to obtain or comply with such licenses." + nl + nl + nl + "OWNERSHIP OF LICENSED SOFTWARE AND CONFIDENTIALITY:" + nl + nl + "7. Ownership." + nl + "Customer acknowledges and agrees that the Licensed Software is and at all times shall remain the exclusive property of HCL TECHNOLOGIES, and that no rights, title or ownership interests of any kind whatsoever in such Licensed Software or any intellectual property rights therein shall pass to Customer." + nl + nl + "8 Notices." + nl + "Customer shall ensure that all proprietary notices of HCL TECHNOLOGIES remain on the Licensed Software in any form." + nl + nl + nl + "WARRANTIES AND INDEMNITIES" + nl + nl + "9. Limitation of Warranties and Liability." + nl + " HCL TECHNOLOGIES warrants that it has the right to grant the license to the Licensed Software GRANTED herein. EXCEPT AS OTHERWISE EXPRESSLY STATED IN THIS AGREEMENT, THERE ARE NO

EXPRESS OR IMPLIED WARRANTIES OR CONDITIONS IN RELATION TO ANY SOFTWARE, DOCUMENTATION, SUPPORT, SERVICES OR PRODUCTS THAT ARE THE SUBJECT MATTER OF THIS AGREEMENT INCLUDING ANY IMPLIED WARRANTIES OR CONDITIONS OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE AND THOSE OTHERWISE ARISING BY STATUTE OR OTHERWISE IN LAW, OR FROM A COURSE OF DEALING OR USAGE OF TRADE. NEITHER HCL TECHNOLOGIES NOR ITS SUPPLIERS, AGENTS, AUTHORIZED RESELLERS, EMPLOYEES, OFFICERS AND DIRECTORS SHALL HAVE ANY LIABILITY TO CUSTOMER OR ANY OTHER PERSON OR ENTITY FOR ANY INDIRECT, INCIDENTAL, SPECIAL OR CONSEQUENTIAL DAMAGES WHATSOEVER; OR FOR LOSS OF REVENUE OR PROFIT, FAILURE TO REALIZE ANTICIPATED PROFITS OR SAVINGS, LOST OR DAMAGED DATA; OR FOR COMMERCIAL OR ECONOMIC LOSS; OR FOR CLAIMS OF A THIRD PARTY; EVEN IF HCL TECHNOLOGIES HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES OR THEY ARE FORESEEABLE. IN ANY EVENT, In respect of any claim, demand or action arising out of this Agreement against HCL TECHNOLOGIES or any of its employees, directors, officers, agents OR AUTHORIZED RESELLERS whether based in contract, tort (including negligence), or otherwise, including a breach by HCL TECHNOLOGIES of any of its obligations under this

Agreement (whether or not a fundamental breach), CUSTOMER SHALL BE LIMITED TO RECEIVING actual and direct damages to a maximum aggregate amount equal to THE AMOUNTS THEN PAID BY CUSTOMER TO HCL TECHNOLOGIES OR ITS AUTHORIZED RESELLER HEREUNDER." + nl + nl + "10. Customer's Intellectual Property Indemnity." + nl + "Customer shall defend at its own expense any claim, proceeding or suit (hereinafter referred to in this Section as a "Claim") brought against HCL TECHNOLOGIES to the extent such Claim alleges that versions of the Licensed Software that have been altered or modified by Customer or on its behalf by a third party, including modifications permitted under Section 5 hereof, or the combination, operation or use of the Licensed Software with non-HCL TECHNOLOGIES software, (provided such claim would have been avoided by the use of the unaltered or non-combination version thereof in accordance with the terms hereof), infringes or misappropriates an intellectual property right of a third party, and will pay all damages which by final judgment or settlement may be assessed against HCL TECHNOLOGIES on account of such infringement or misappropriation together with all costs and expenses (including reasonable legal fees) incurred by HCL TECHNOLOGIES as a direct result thereof, provided that Customer is given: (i) prompt written notice of the Claim or of any allegations or circumstances known to HCL TECHNOLOGIES which could result in a Claim; (ii) all reasonable information and assistance which it may require

to defend the Claim; and (iii) sole control of the defence of the Claim, and all negotiations for its settlement or compromise thereof, provided that HCL TECHNOLOGIES is entitled to participate in the defence and any negotiations at its own expense. Customer agrees not to use the Licensed Software to infringe the copyright of third parties. Customer agrees to indemnify HCL TECHNOLOGIES for all damages, costs and expenses (including reasonable legal fees) incurred by HCL TECHNOLOGIES as a result of any claim, proceeding or suit against HCL TECHNOLOGIES alleging that Customer used the Licensed Software to infringe the copyright of a third party." + nl + nl + nl + "GENERAL" + nl + nl + "12. General." + nl + "Except in the case of HCL TECHNOLOGIES authorized resellers who have also entered into a Business Partner Reseller Agreement with HCL TECHNOLOGIES, this Agreement is the entire agreement between the parties in relation to any software, documentation, support, services, or products that are the subject matter of this Agreement and supersedes any other agreements or discussions, oral or written. This Agreement may not be changed except by a written amendment signed by Customer and HCL TECHNOLOGIES. No provision of this Agreement may be waived except in writing signed by the party giving the waiver. Only a signing officer of HCL TECHNOLOGIES has the authority to change or waive this agreement. Customer may not assign this Agreement. This Agreement shall be governed and construed in accordance with the laws of the Province of Ontario, Canada. The parties

expressly exclude the application of the United Nations Convention on Contracts for International Sale of Goods. The parties have requested that this Agreement and all documents contemplated hereby be drawn up in English. Les parties aux présentes ont exigé que cette entente et tous autres documents envisagés par les présentes soient rédigés en anglais. If any provision of this Agreement is declared by a Court of competent jurisdiction to be invalid, illegal or "unenforceable, such provision shall be severed from the Agreement and the other provisions shall remain in full force and effect. In this Agreement, "including" means "including without limitation" and is not to be construed to limit any general statement which it follows to the specific or similar items or matters immediately following it. Customer hereunder shall pay all amounts due within thirty (30) days of the date of an invoice for such amounts. If Customer fails to pay any amount when due, HCL TECHNOLOGIES or its authorized reseller shall have the right, in addition to any other remedies, to charge, and Customer shall pay, interest on such overdue amounts at the rate of one and a half percent (1.5%) per month (18% per annum). In addition to all charges hereunder, Customer shall pay to HCL TECHNOLOGIES or its authorized reseller all taxes, duties, and other such assessments or charges which may be assessed, levied or imposed with respect to any software, support, services or products provided hereunder, except taxes based on HCL TECHNOLOGIES or its authorized reseller's income or capital. This Agreement may be

executed in any number of counterparts including by Customer signing and returning the first page hereof to HCL TECHNOLOGIES by facsimile transmission for signature by HCL TECHNOLOGIES. Each executed counterpart shall be deemed to be an original; all executed counterparts taken together shall constitute one agreement." + nl + nl + "13. Transfer of Information" + nl + "During the installation, you agree to provide HCL TECHNOLOGIES with information regarding the installation process. The information provided will be used for internal audit purposes only to ensure the correct amount of licenses are enabled within the application." + nl + nl + "14. Restricted Rights." + nl + "If this Licensed Software is acquired by or for the U.S. Government then it is provided with Restricted Rights. Use, duplication, or disclosure by the U.S. Government is subject to restrictions under applicable law, including those set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013, or subparagraphs (c)(1) and (2) of the Commercial Computer Software - Restricted Rights at 48 CFR 52.227-19, as applicable. Contractor/manufacturer is HCL TECHNOLOGIES," + nl + nl + nl + "Survival. Sections 3 to 6, 9 to 11, 13 and 14 shall survive any termination of this Agreement";

Is Scripted:

Yes

Hotspot

Type:

Caption:

Hide When:

JavaScript & HTML Code:

[None]

LotusScript Code:

(Field) License

Button

Continue

Accept = "1";

```

Sub Entering(Source As Field)
    Dim wk As New NotesUIWorkspace
    Call wk.CurrentDocument.GotoField("Untitled")
End Sub
(Button) Continue
Sub Click(Source As Button)
    Dim wk As New NotesUIWorkspace
    Dim doc As NotesDocument
    Set doc = wk.CurrentDocument.Document
    If doc.Accept(0)="2" Then Exit Sub
%REM
    Dim session As New NotesSession
    Dim db As NotesDatabase
    Dim memo As NotesDocument
    Set db = session.CurrentDatabase
    Set memo = New NotesDocument(db)
    memo.Subject = db.Title + " - LICENSE AGREEMENT TERMS AND CONDITIONS"
    memo.Body = db.Title + " - LICENSE AGREEMENT TERMS AND CONDITIONS" + Chr(10)+Chr(10)_
    + doc.License(0)
    Call memo.Send(False,session.UserName)
%END REM
    Call wk.RefreshParentNote
    Call wk.CurrentDocument.Close(True)

```

Name:

Alias:
 Last Modification:
 Comment:
 Type:
 Window Title Formula:

Include in Compose Menu:
 Include in Query by Form:
 Default Database Form:
 Automatically Refresh Fields:
 Mail New Documents When Saving:
 Store Form In Documents:
 Inherit Default Field Values:
 Updates Become Responses:
 Retain Prior Versions As Responses:
 Activate Objects When Composing:
 Activate Objects When Editing:
 Activate Objects When Reading:
 Document Encryption Keys:
 Composed Documents May Be Read By:
 Form May Be Composed By:

fmGroup

Group
 01/11/2023 02:55:14 PM
 [Not Assigned]
 Document
 "Deny Access Group
 Information"
 No
 No
 No
 No
 No
 No
 No
 No
 No
 No
 No
 [None Assigned]
 All Users
 All Users

Subcomponents:

Field:
 Datatype:
 Time Format:
 Help Description:
 Field Type:
 Sign When Mailing/In Section:
 Encryption:
 Update Requires Editor Access:
 Formula:

NABNOTEUPDATED
 Time
 01/25/2023 11:01 AM
 [Not Assigned]
 Computed when composed
 No
 Disabled
 No
 @ThisValue

Is Scripted:	No
Action:	Spacer
Action:	FormName
Action:	Spacer
Action:	Close
Formula:	@Command([CloseWindow]);
Action:	Print
Formula:	@Command([FilePrint]);
JavaScript & HTML Code:	
[None]	
LotusScript Code:	
(Form) Group	
Sub Queryopen(Source As Notesuidocument, Mode As Integer, Isnewdoc As Variant, Continue As Variant)	
If source.EditMode Then	
Print "Document Edits are not allowed"	
Continue=False	
End If	
End Sub	
Sub Querymodechange(Source As Notesuidocument, Continue As Variant)	
If Not source.EditMode Then	
Print "Document Edits are not allowed"	
Continue=False	
End If	
End Sub	
[None]	
Name:	fmHistory
Alias:	History
Last Modification:	01/11/2023 02:55:17 PM
Comment:	[Not Assigned]
Type:	Document
Window Title Formula:	"HCL Domino License Analysis - Results From " + @Text(@Created; "D0T1Z0");
Include in Compose Menu:	No
Include in Query by Form:	No
Default Database Form:	No
Automatically Refresh Fields:	No
Mail New Documents When Saving:	No
Store Form In Documents:	No
Inherit Default Field Values:	No
Updates Become Responses:	No
Retain Prior Versions As Responses:	No
Activate Objects When Composing:	No
Activate Objects When Editing:	No
Activate Objects When Reading:	No
Document Encryption Keys:	[None Assigned]
Composed Documents May Be Read By:	All Users
Form May Be Composed By:	All Users
Subcomponents:	
Field:	SaveOptions
Datatype:	Text
Help Description:	[Not Assigned]
Field Type:	Computed
Sign When Mailing/In Section:	No
Encryption:	Disabled

Update Requires Editor Access:	No
Formula:	"0"
Is Scripted:	No
Field:	PersonDocCount
Datatype:	Number
Number Format:	General
Percentage (value * 100)%:	No
Parentheses on Negative Numbers:	No
Punctuated at Thousands:	No
Help Description:	[Not Assigned]
Field Type:	Computed when composed
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Formula:	@ThisValue
Is Scripted:	No
Field:	GroupDocCount
Datatype:	Number
Number Format:	General
Percentage (value * 100)%:	No
Parentheses on Negative Numbers:	No
Punctuated at Thousands:	No
Help Description:	[Not Assigned]
Field Type:	Computed when composed
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Formula:	val := @DbLookup("" : "NoCache"; "" ; "config"; "config"; @ThisName);@If(@IsError(val); -20; @IsNumber(val); val; -1);
Is Scripted:	No
Field:	DAUserCount
Datatype:	Number
Number Format:	General
Percentage (value * 100)%:	No
Parentheses on Negative Numbers:	No
Punctuated at Thousands:	No
Help Description:	[Not Assigned]
Field Type:	Computed when composed
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Formula:	@ThisValue
Is Scripted:	No
Field:	LicenseTrackingDocCount
Datatype:	Number
Number Format:	General
Percentage (value * 100)%:	No
Parentheses on Negative Numbers:	No
Punctuated at Thousands:	No
Help Description:	[Not Assigned]
Field Type:	Computed when composed
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No

Formula:	@ThisValue
Is Scripted:	No
Field:	ValidUserCount
Datatype:	Number
Number Format:	General
Percentage (value * 100)%:	No
Parentheses on Negative Numbers:	No
Punctuated at Thousands:	No
Help Description:	[Not Assigned]
Field Type:	Computed when composed
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Formula:	@ThisValue
Is Scripted:	No
Field:	EntitlementTrackingServers
Datatype:	Text
Input Multi-Value Separator(s):	Comma, Semicolon, New Line
Display Multi-Value Separator:	New Line
Help Description:	[Not Assigned]
Field Type:	Computed when composed
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Formula:	@ThisValue
Is Scripted:	No
Field:	UserReportLocation
Datatype:	Text
Help Description:	[Not Assigned]
Field Type:	Computed when composed
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Formula:	@ThisValue
Is Scripted:	No
Field:	DAScanError
Datatype:	Text
Help Description:	[Not Assigned]
Field Type:	Computed when composed
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Formula:	@ThisValue
Is Scripted:	No
Field:	LicenseTrackingServersMissing
Datatype:	Text
Input Multi-Value Separator(s):	Comma, Semicolon, New Line
Display Multi-Value Separator:	New Line
Help Description:	[Not Assigned]
Field Type:	Computed when composed
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Formula:	@ThisValue

Is Scripted:	No
Field:	AdminServer
Datatype:	Text
Help Description:	[Not Assigned]
Field Type:	Computed when composed
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Formula:	@ThisValue
Is Scripted:	No
Field:	ServerListDisp
Datatype:	Keywords
Help Description:	[Not Assigned]
Field Type:	Editable
Keyword User Interface:	Standard
Allow Values Not In List:	No
Allowable Keywords Formula:	val := ServerList; @If(@IsError(val); "< Error Retrieving Server Names >"; @Word(val; " "; 1) + " (" + @Word(val; " "; 2) + ")");
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Default Value Formula:	""
Is Scripted:	No
Field:	EntitlementTrackingServersDisp
Datatype:	Keywords
Help Description:	[Not Assigned]
Field Type:	Editable
Keyword User Interface:	Standard
Allow Values Not In List:	No
Allowable Keywords Formula:	EntitlementTrackingServers
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Is Scripted:	No
Field:	DALInfo
Datatype:	Text
Help Description:	[Not Assigned]
Field Type:	Computed when composed
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Formula:	val1 := DAName;val2 := DAFilename; @If(@IsError(val1); "< Error Retrieving Directory Assistance Information >"; val1 = ""; "Directory Assistance is not configured."; val1 + " (" + val2 + ")");
Is Scripted:	No
Field:	DAListExternalDisp

Datatype:	Text
Help Description:	[Not Assigned]
Field Type:	Computed when composed
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Formula:	val := DAListExternal; @If(@IsError(val); "< Error Retrieving Server Names >"; @Word(val; "~"; 1));
Is Scripted:	No
Hide When:	DAListExternal = "";
Field:	DAListDisp
Datatype:	Text
Input Multi-Value Separator(s):	Comma, Semicolon, New Line
Display Multi-Value Separator:	New Line
Help Description:	[Not Assigned]
Field Type:	Computed when composed
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Formula:	val := DAList;ret := @ReplaceSubstring(@If(@IsErr or(val); "< Error Retrieving Server Names >"; @Word(val; "~"; 1)); " "; "!!"); @Trim(@If(DAListExternal = ""; ret; @Replace(ret; DAListExternal; "")));
Is Scripted:	No
Hide When:	DALInfo = "";
Field:	ObservationsDisp
Datatype:	Text
Help Description:	[Not Assigned]
Field Type:	Computed for display
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Formula:	val := Observations; @If(@IsError(val); ""; @Implode("- " + @Word(val; " "; 1); @NewLine));
Is Scripted:	No
Subform:	sfrmInfoOverlays
Action:	Spacer
Action:	FormName
Action:	Spacer
Action:	Close
Formula:	@Command([CloseWindow]);
Action:	Print
Formula:	@Command([FilePrint]);
JavaScript & HTML Code:	
Field LicenseTrackingServersMissing	
onClick	
onClick	
onClick	

LotusScript Code: (Globals)

Name:

Subcomponents:

[illegible]

Field:

Datatype:
 Help Description:
 Field Type:
 Sign When Mailing/In Section:
 Encryption:
 Update Requires Editor Access:
 Default Value Formula:
 Is Scripted:

Field:

Datatype:
 Help Description:
 Field Type:
 Sign When Mailing/In Section:
 Encryption:
 Update Requires Editor Access:
 Default Value Formula:
 Is Scripted:

Field:

Datatype:
 Help Description:
 Field Type:
 Sign When Mailing/In Section:
 Encryption:
 Update Requires Editor Access:
 Formula:
 Is Scripted:

Field:

Datatype:
 Help Description:
 Field Type:
 Sign When Mailing/In Section:
 Encryption:
 Update Requires Editor Access:
 Formula:

Is Scripted:

Field:

Datatype:
 Number Format:
 Percentage (value * 100)%:
 Parentheses on Negative Numbers:
 Punctuated at Thousands:
 Help Description:
 Field Type:
 Sign When Mailing/In Section:
 Encryption:
 Update Requires Editor Access:
 Formula:

\$Buttons

Text
 [Not Assigned]
 Editable
 No
 Disabled
 No
 "1"
 No

\$Steps

Text
 [Not Assigned]
 Editable
 No
 Disabled
 No
 "1"
 No

SaveOptions

Text
 [Not Assigned]
 Computed when composed
 No
 Disabled
 No
 "0"
 No

ConfigCheck

Text
 [Not Assigned]
 Computed when composed
 No
 Disabled
 No
 val := @DbLookup("":
 "NoCache"; @DbName;
 "config"; "config";
 "ScansComplete");
 @If(@IsError(val); ""; val = ""; "";
 val);
 No

PersonDocCount

Number
 General
 No
 No
 No
 [Not Assigned]
 Computed
 No
 Disabled
 No
 val := @DbLookup("":
 "NoCache"; ""; "config"; "config";
 @ThisName);@If(@IsError(val);
 -20; @IsNumber(val); val; -1);

Is Scripted:	No
Field:	GroupDocCount
Datatype:	Number
Number Format:	General
Percentage (value * 100)%:	No
Parentheses on Negative Numbers:	No
Punctuated at Thousands:	No
Help Description:	[Not Assigned]
Field Type:	Computed
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Formula:	val := @DbLookup("" : "NoCache"; "" ; "config"; "config"; @ThisName);@If(@IsError(val); -20; @IsNumber(val); val; -1);
Is Scripted:	No
Field:	DAUserCount
Datatype:	Number
Number Format:	General
Percentage (value * 100)%:	No
Parentheses on Negative Numbers:	No
Punctuated at Thousands:	No
Help Description:	[Not Assigned]
Field Type:	Computed
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Formula:	val := @DbLookup("" : "NoCache"; "" ; "config"; "config"; @ThisName);@If(@IsError(val); -20; @IsNumber(val); val; -1);
Is Scripted:	No
Field:	LicenseTrackingDocCount
Datatype:	Number
Number Format:	General
Percentage (value * 100)%:	No
Parentheses on Negative Numbers:	No
Punctuated at Thousands:	No
Help Description:	[Not Assigned]
Field Type:	Computed
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Formula:	val := @DbLookup("" : "NoCache"; "" ; "config"; "config"; @ThisName);@If(@IsError(val); -20; @IsNumber(val); val; -1);
Is Scripted:	No
Field:	ValidUserCount
Datatype:	Number
Number Format:	General
Percentage (value * 100)%:	No
Parentheses on Negative Numbers:	No
Punctuated at Thousands:	No
Help Description:	[Not Assigned]

Field Type:	Computed
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Formula:	val := @DbLookup("": "NoCache"; "", "config"; "config"; @ThisName);@If(@IsError(val); -10101; val);
Is Scripted:	No
Field:	EntitlementTrackingServers
Datatype:	Text
Input Multi-Value Separator(s):	Comma, Semicolon, New Line
Display Multi-Value Separator:	New Line
Help Description:	[Not Assigned]
Field Type:	Computed
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Formula:	val := @DbLookup("": "NoCache"; "", "config"; "config"; @ThisName);@If(@IsError(val); "< Error Retrieving Server Names >"; @Word(val; " "; 1));
Is Scripted:	No
Field:	UserReportLocation
Datatype:	Text
Help Description:	[Not Assigned]
Field Type:	Computed
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Formula:	val := @DbLookup("": "NoCache"; "", "config"; "config"; @ThisName);@If(@IsError(val); "< Error Retrieving Server Names >"; @Word(val; " "; 1));
Is Scripted:	No
Field:	DAScanError
Datatype:	Text
Help Description:	[Not Assigned]
Field Type:	Computed
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Formula:	check := @DbLookup("": "NoCache"; "", "config"; "config"; "DAScanError");@If(@IsError(check); "", "Note: there was an error scanning the additional Domino Directories");
Is Scripted:	No
Field:	LicenseTrackingServersMissing
Datatype:	Text

Input Multi-Value Separator(s):	Comma, Semicolon, New Line
Display Multi-Value Separator:	New Line
Help Description:	[Not Assigned]
Field Type:	Computed when composed
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Formula:	val := @DbLookup("" : "NoCache"; "" ; "config"; "config"; @ThisName);@If(@IsError(val); "< Error Retrieving Server Names >"; val);
Is Scripted:	No
Hotspot	
Type:	Button
Caption:	PerformScan
Formula:	scanType := @If(Step = 2; "Config"; Step = 3; "User"; Step = 4; "DAGroups"; "Entitlement"); FIELD Scan := scanType; @Command([RefreshHideForm ulas]); @PostedCommand([ToolsRun Macro]; "PullInfo" + scanType);
Hotspot	
Type:	Button
Caption:	MakeHistory
Formula:	@Command([ToolsRunMacro]; "MakeHistory");
Hotspot	
Type:	Button
Caption:	FirstNext
Is Scripted:	Yes, see below
Field:	Step
Datatype:	Number
Number Format:	General
Percentage (value * 100)%:	No
Parentheses on Negative Numbers:	No
Punctuated at Thousands:	No
Help Description:	[Not Assigned]
Field Type:	Editable
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Default Value Formula:	1
Is Scripted:	No
Field:	Scan
Datatype:	Text
Help Description:	[Not Assigned]
Field Type:	Editable
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Default Value Formula:	@ThisValue
Is Scripted:	No
Field:	UserScan

Datatype:	Text
Help Description:	[Not Assigned]
Field Type:	Editable
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Default Value Formula:	@ThisValue
Is Scripted:	No
Field:	DAScan
Datatype:	Text
Help Description:	[Not Assigned]
Field Type:	Editable
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Default Value Formula:	@ThisValue
Is Scripted:	No
Field:	EntitlementScan
Datatype:	Text
Help Description:	[Not Assigned]
Field Type:	Editable
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Default Value Formula:	@ThisValue
Is Scripted:	No
Hotspot	
Type:	Link
Caption:	HCL Domino CCB License Agreement
Associated Data:	https://www.hcltechsw.com/wps/wcm/connect/c6f1e7e3-d6d3-436d-a95a-44c5347dca1a/HCL+Domino+Complete+Collaboration+12.0.1.pdf?MOD=AJPERES&CONVERT_TO=url&CACHEID=ROOTWORKSPACE-c6f1e7e3-d6d3-436d-a95a-44c5347dca1a-nSOi.Tg
Hotspot	
Type:	Link
Caption:	User Report Tool
Associated Data:	https://support.hcltechsw.com/csm?id=kb_article&sysparm_article=KB0095328
Hotspot	
Type:	Link
Caption:	HCL Domino Administration Documentation
Associated Data:	https://help.hcltechsw.com/domino/12.0.0/admin/index.html
Hotspot	
Type:	Link

Help Description:	[Not Assigned]
Field Type:	Computed
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Formula:	<pre>val := @DbLookup("" : "NoCache"; "" ; "config"; "config"; @ThisName);@If(@IsError(val); "< Error Retrieving Server Names >"; @Word(val; "~"; 1)); No DAListExternal = "" AdminServer = "";</pre>
Is Scripted:	DAList
Hide When:	Text
Field:	Comma, Semicolon, New Line
Datatype:	New Line
Input Multi-Value Separator(s):	[Not Assigned]
Display Multi-Value Separator:	Computed
Help Description:	No
Field Type:	Disabled
Sign When Mailing/In Section:	No
Encryption:	No
Update Requires Editor Access:	<pre>val := @DbLookup("" : "NoCache"; "" ; "config"; "config"; "DAList");ret := @ReplaceSubstring(@If(@IsErr or(val); "< Error Retrieving Server Names >"; @Word(val; ~"; 1)); " "; "!!"); @Trim(@If(DAListExternal = "" ; ret; @Replace(ret; DAListExternal; ""))); No DALInfo = "" AdminServer = "";</pre>
Formula:	
Is Scripted:	Action Hotspot
Hide When:	Start Users Scan Now
Hotspot	
Type:	
Caption:	
Hotspot	
Type:	Action Hotspot
Caption:	Person Documents
Formula:	<pre>@Do(@SetTargetFrame("Sideb ar"); @Command([OpenFrameset]; "fsSidebar2")); @Do(@SetTargetFrame("Main"); @PostedCommand([OpenView] ; "UserInfo"));</pre>
Hotspot	
Type:	Action Hotspot
Caption:	Start Deny Access Scan Now
Hotspot	
Type:	Action Hotspot
Caption:	Deny Access Groups
Formula:	

Hotspot

Type:
Caption:
Formula:

```
@Do(@SetTargetFrame("Sideb  
ar");  
@Command([OpenFrameset];  
"fsSidebar3"));  
@Do(@SetTargetFrame("Main"  
);  
@PostedCommand([OpenView]  
; "dagroups"));
```

Action Hotspot
Deny Access Users Found

```
@Do(@SetTargetFrame("Sideb  
ar");  
@Command([OpenFrameset];  
"fsSidebar3"));  
@Do(@SetTargetFrame("Main"  
);  
@PostedCommand([OpenView]  
; "dagroups"));
```

Action Hotspot
Start Entitlement Tracking Scan
Now

Hotspot

Type:
Caption:

Field:

Datatype:
Help Description:
Field Type:
Keyword User Interface:
Allow Values Not In List:
Allowable Keywords Formula:
Sign When Mailing/In Section:
Encryption:
Update Requires Editor Access:
Is Scripted:

**EntitlementTrackingServersDis
p**

Keywords
[Not Assigned]
Editable
Standard
No
EntitlementTrackingServers
No
Disabled
No
No

Hotspot

Type:
Caption:

Action Hotspot
Start User Report Tool Scan
Now

Hotspot

Type:
Caption:
Formula:

Action Hotspot
Show Previous Results
@SetTargetFrame("Inner");
@Command([OpenFrameset];
"fsInner");

Hotspot

Type:
Caption:

Action Hotspot
Next

Hotspot

Type:
Caption:
Formula:

Action Hotspot
Cancel
@Command([CloseWindow]);

Hotspot

Type:

Action Hotspot

Caption:	Back
Is Scripted:	Yes, see below
Hotspot	
Type:	Action Hotspot
Caption:	Next
Is Scripted:	Yes, see below
Hotspot	
Type:	Action Hotspot
Caption:	Cancel
Formula:	@Command([CloseWindow]);
Hotspot	
Type:	Action Hotspot
Caption:	Back
Is Scripted:	Yes, see below
Hotspot	
Type:	Action Hotspot
Caption:	Next
Is Scripted:	Yes, see below
Hotspot	
Type:	Action Hotspot
Caption:	Cancel
Formula:	@Command([CloseWindow]);
Hotspot	
Type:	Action Hotspot
Caption:	Back
Is Scripted:	Yes, see below
Hotspot	
Type:	Action Hotspot
Caption:	Next
Is Scripted:	Yes, see below
Hotspot	
Type:	Action Hotspot
Caption:	Cancel
Formula:	@Command([CloseWindow]);
Hotspot	
Type:	Action Hotspot
Caption:	Back
Is Scripted:	Yes, see below
Hotspot	
Type:	Action Hotspot
Caption:	Next
Is Scripted:	Yes, see below
Hotspot	
Type:	Action Hotspot
Caption:	Cancel
Formula:	@Command([CloseWindow]);
Hotspot	
Type:	Action Hotspot
Caption:	Back
Is Scripted:	Yes, see below
Hotspot	
Type:	Action Hotspot
Caption:	Next
Is Scripted:	Yes, see below
Hotspot	
Type:	Action Hotspot

Caption:
Formula:
Hotspot
Type:
Caption:
Formula:

Subform:
JavaScript & HTML Code:

Action Hotspot
Client onClick
document.forms[0].Timer.click();
setTimeout(document.forms[0].PerformScan.click(), 1);
onClick
Field AdminServer
onClick
Field ServerList
onClick
onClick
Action Hotspot
Client onClick
document.forms[0].Timer.click();
setTimeout(document.forms[0].PerformScan.click(), 1);
onClick
Action Hotspot
Client onClick
document.forms[0].Timer.click();
setTimeout(document.forms[0].PerformScan.click(), 1);
onClick
onClick
Action Hotspot
Client onClick
document.forms[0].Timer.click();
setTimeout(document.forms[0].PerformScan.click(), 1);
onClick
onClick
Action Hotspot
Client onClick
document.forms[0].Timer.click();
setTimeout(document.forms[0].PerformScan.click(), 1);
onClick
onClick
Action Hotspot
Client onClick
document.forms[0].MakeHistory.click();
document.forms[0].FirstNext.click()

LotusScript Code:

(Globals)
Option Public
Option Declare
Use "slCore"
Dim stepVal As String
Dim elapsedTime As NotesTimer
Dim timerSeconds As Integer
Dim checkStep As Integer
Dim processComplete As Boolean
Dim openWS As NotesUIWorkspace

Cancel
@Command([CloseWindow]);

Action Hotspot
View Results
@SetTargetFrame("Inner");
@Command([OpenFrameset];
"fsInner");
sfmlInfoOverlays

```

Dim openUI As NotesUIDocument
Dim openNote As NotesDocument
Dim openSession As NotesSession
Dim openDB As NotesDatabase
Dim openView As NotesView
Dim openAdmin As NotesDocument
Sub checkSettings (Source As NotesTimer)
    On Error Goto errHandle
    Set openAdmin=Nothing
    Set openSession=New NotesSession
    Set openDB=openSession.CurrentDatabase
    If openGetAdminValue("Critical")(0)<>"" Then Call openUI.Close
    If openGetAdminValue("PauseTime")(0)<>"" Then Exit Sub
    If checkStep=0 Then
        If openGetAdminValue("ConfigPullStatus")(0)="Complete" Then
            checkStep=1
            openNote.AdminServer=openGetAdminValue("AdminServer")(0)
            openNote.Scan=""
            openSetAdminValue "NABPersonDocSyncStatus", ""
            Call openUI.Refresh()
            Set source=Nothing
        End If
    ElseIf checkStep=1 Then
        If openGetAdminValue("NABPersonDocSyncStatus")(0)="Complete" Then
            checkStep=2
            openNote.UserScan="complete"
            openNote.Scan=""
            openSetAdminValue "DADocSyncStatus", ""
            Call openUI.Refresh()
            Set source=Nothing
        End If
    ElseIf checkStep=2 Then
        If openGetAdminValue("DADocSyncStatus")(0)="Complete" Then
            checkStep=3
            openNote.DAScan="complete"
            openNote.Scan=""
            openSetAdminValue "EntitlementSyncStatus", ""
            Call openUI.Refresh()
            Set source=Nothing
        End If
    Else
        If openGetAdminValue("EntitlementSyncStatus")(0)="Complete" Then
            openNote.EntitlementScan="complete"
            openNote.Scan=""
            Call openUI.Refresh()
            Set source=Nothing
        End If
        openSetAdminValue "ScansComplete", "true"
    End If
errHandle:
    Err=0
    Exit Sub
End Sub
Function openGetAdminValue(fldName As String) As Variant
    ' On Error Goto errHandle
    Dim empty(0) As String

```



```

    If openAdmin Is Nothing Then
        Set openView=openDB.Getview("config")
        Set openAdmin=openView.Getdocumentbykey("config",True)
        If openAdmin Is Nothing Then
            Set openAdmin=openDb.Createdocument()
            openAdmin.Form="config"
            Call openAdmin.Save(True,True,True)
        End If
    End If
    openGetAdminValue=empty
    If openAdmin.Hasitem(fldName) Then openGetAdminValue=openAdmin.Getfirstitem(fldName).Values
errHandle:
'    If Err<>0 Then MsgBox "ERROR in openGetAdminValue: "+Error+Chr(10)+"#: "+Cstr(Err)+"", line: "+Cstr(E
Exit Function
End Function
Sub openSetAdminValue (fldName As String,value As Variant)
    If openadmin Is Nothing Then
        Set openview=opendb.Getview("config")
        Set openadmin=openview.Getdocumentbykey("config",t)
        If openadmin Is Nothing Then
            Set openadmin=opendb.Createdocument()
            openadmin.Form="config"
        End If
    End If
    Call openadmin.Removeitem(fldName)
    Call openadmin.Replaceitemvalue(fldName,value)
    Call openadmin.Save(t,t,t)
End Sub
Sub CreateDigitalSignature()
    Dim encStr As String
    Dim adminVal As Variant
    Dim nl As String
    nl=Chr(10)
    encStr=|"Date:"|+Cstr(Now)+|",|+nl
    encStr=encStr+|"User:"|+openSession.Username+|",|+nl
    adminVal=openGetAdminValue("ValidUserCount")
    If Cstr(adminVal(0))="" Then encStr=encStr+|"AU":"0",|+nl Else encStr=encStr+|"AU:"|+Cstr(adminVal(0))-
    adminVal=openGetAdminValue("LDAPDA")
    If Cstr(adminVal(0))<>"" Then encStr=encStr+|"LDAP":"LDAP Directory In Use at this organization",|+nl
    adminVal=openGetAdminValue("PersonDocCount")
    If Cstr(adminVal(0))="" Then encStr=encStr+|"PD":"0",|+nl Else encStr=encStr+|"PD:"|+Cstr(adminVal(0))-
    adminVal=openGetAdminValue("MailUserCount")
    If Cstr(adminVal(0))="" Then encStr=encStr+|"MU":"0",|+nl Else encStr=encStr+|"MU:"|+Cstr(adminVal(0))-
    adminVal=openGetAdminValue("GroupDocCount")
    If Cstr(adminVal(0))="" Then encStr=encStr+|"GD":"0",|+nl Else encStr=encStr+|"GD:"|+Cstr(adminVal(0))-
    adminVal=openGetAdminValue("DAUserCount")
    If Cstr(adminVal(0))="" Then encStr=encStr+|"DAU":"0",|+nl Else encStr=encStr+|"DAU:"|+Cstr(adminVal(0))-
    adminVal=openGetAdminValue("LicenseTrackingDocCount")
    If Cstr(adminVal(0))="" Then encStr=encStr+|"LTD":"0"| Else encStr=encStr+|"LTD:"|+Cstr(adminVal(0))+
    encStr=EncodeBase64(encStr)
    encStr=EncodeBase64(encStr)
    openSetAdminValue "Signature",encStr
    openSetAdminValue "SignatureDate",Now
    openSetAdminValue "SignatureUser",openSession.Username
End Sub
Function EncodeBase64 (StrIn As String) As String

```

```

    On Error Goto errHandle
    Dim stream As NotesStream
    Dim tmpDoc As NotesDocument
    Dim body As NotesMIMEEntity
    Set stream = openSession.CreateStream
    Call stream.WriteText (StrIn)
    Set opendb = openSession.CurrentDatabase
    Set tmpDoc = opendb.CreateDocument
    Set body = tmpDoc.CreateMIMEEntity
    Call body.SetContentFromText (stream, "", ENC_NONE)
    Call body.EncodeContent (ENC_BASE64)
    EncodeBase64 = Replace(Replace(body.ContentAsText, Chr(13),""), Chr(10), "")
errHandle:
    Call stream.Close
    Set tmpdoc = Nothing
    If Err <> 0 Then MsgBox "error while creating digital signature"+Chr(10)+"error: "+Error+Chr(10)+"line: "+Cs
    Exit Function
End Function
(Form) fmLaunch
Sub Postopen(Source As Notesuidocument)
    Set openWS=New NotesUIWorkspace
    Set openUI=source
    Set openNote=source.Document
End Sub
Sub Queryclose(Source As Notesuidocument, Continue As Variant)
    If source.Document.Step(0)=7 Then openSetAdminValue "AllScansComplete","true"
End Sub
Sub Queryopen(Source As Notesuidocument, Mode As Integer, Isnewdoc As Variant, Continue As Variant)
    Set openSession=New NotesSession
    Set openDB=openSession.CurrentDatabase
    Set openView=openDB.Getview("config")
    Set openAdmin=openView.Getdocumentbykey("config",t)
    If Not openAdmin Is Nothing Then
        If openAdmin.HasItem("Critical") Then
            Call openAdmin.RemoveItem("Critical")
            Call openAdmin.Save(t,t,t)
        End If
    End If
End Sub
(Button) Timer
Sub Click(Source As Button)
    timerSeconds=5
    Set elapsedTime = New NotesTimer(timerSeconds, "")
    On Event Alarm From elapsedTime Call checkSettings
End Sub
(Button) FirstNext
Sub Click(Source As Button)
    stepVal="2"
    opennote.~$Buttons =stepVal
    opennote.~$Steps =stepVal
    opennote.Step=Cint(stepVal)
    Call openui.Refresh
End Sub
(Hotspot) Back
Sub Click(Source As Button)
    stepVal="1"

```

```

        opennote.~$Buttons =stepVal
        opennote.~$Steps =stepVal
        opennote.Step=Cint(stepVal)
        Call openui.Refresh
    End Sub
(Hotspot) Next
    Sub Click(Source As Button)
        stepVal="3"
        opennote.~$Buttons =stepVal
        opennote.~$Steps =stepVal
        opennote.Step=Cint(stepVal)
        Call openui.Refresh
    End Sub
(Hotspot) Back
    Sub Click(Source As Button)
        stepVal="2"
        opennote.~$Buttons =stepVal
        opennote.~$Steps =stepVal
        opennote.Step=Cint(stepVal)
        Call openui.Refresh
    End Sub
(Hotspot) Next
    Sub Click(Source As Button)
        stepVal="4"
        opennote.~$Buttons =stepVal
        opennote.~$Steps =stepVal
        opennote.Step=Cint(stepVal)
        Call openui.Refresh
    End Sub
(Hotspot) Back
    Sub Click(Source As Button)
        stepVal="3"
        opennote.~$Buttons =stepVal
        opennote.~$Steps =stepVal
        opennote.Step=Cint(stepVal)
        Call openui.Refresh
    End Sub
(Hotspot) Next
    Sub Click(Source As Button)
        If openGetAdminValue("EntitlementTrackingServers")(0)<>"" Then
            stepVal="5"
        ElseIf openGetAdminValue("UserReportLocation")(0)<>"" Then
            stepVal="6"
        Else
            CreateDigitalSignature
            stepVal="7"
        End If
        opennote.~$Buttons =stepVal
        opennote.~$Steps =stepVal
        opennote.Step=Cint(stepVal)
        Call openui.Refresh
    End Sub
(Hotspot) Back
    Sub Click(Source As Button)
        stepVal="4"
        opennote.~$Buttons =stepVal

```

```

        opennote.~$Steps =stepVal
        opennote.Step=Cint(stepVal)
        Call openui.Refresh
    End Sub
(Hotspot)    Next
    Sub Click(Source As Button)
        stepVal="7"
        opennote.~$Buttons =stepVal
        opennote.~$Steps =stepVal
        opennote.Step=Cint(stepVal)
        CreateDigitalSignature
        openSetAdminValue "AllScansComplete","true"
        Call openui.Refresh
    End Sub
(Hotspot)    Back
    Sub Click(Source As Button)
        stepVal="4"
        opennote.~$Buttons =stepVal
        opennote.~$Steps =stepVal
        opennote.Step=Cint(stepVal)
        Call openui.Refresh
    End Sub
(Hotspot)    Next
    Sub Click(Source As Button)
        stepVal="7"
        opennote.~$Buttons =stepVal
        opennote.~$Steps =stepVal
        opennote.Step=Cint(stepVal)
        CreateDigitalSignature
        openSetAdminValue "AllScansComplete","true"
        Call openui.Refresh
    End Sub

```

Name:

Alias:
 Last Modification:
 Comment:
 Type:
 Window Title Formula:

 Include in Compose Menu:
 Include in Query by Form:
 Default Database Form:
 Automatically Refresh Fields:
 Mail New Documents When Saving:
 Store Form In Documents:
 Inherit Default Field Values:
 Updates Become Responses:
 Retain Prior Versions As Responses:
 Activate Objects When Composing:
 Activate Objects When Editing:
 Activate Objects When Reading:
 Document Encryption Keys:
 Composed Documents May Be Read By:
 Form May Be Composed By:

fmLog

config
 01/11/2023 02:55:18 PM
 [Not Assigned]
 Document
 "HCL Domino License Analysis -
 Results From " +
 @Text(@Created; "D0T1Z0");
 No
 No
 No
 No
 No
 No
 No
 No
 No
 No
 No
 No
 [None Assigned]
 All Users
 All Users

Subcomponents:

Field:

Datatype:
Help Description:
Field Type:
Sign When Mailing/In Section:
Encryption:
Update Requires Editor Access:
Formula:
Is Scripted:

Field:

Datatype:
Help Description:
Field Type:
Sign When Mailing/In Section:
Encryption:
Update Requires Editor Access:
Formula:

Is Scripted:

Field:

Datatype:
Help Description:
Field Type:
Sign When Mailing/In Section:
Encryption:
Update Requires Editor Access:
Is Scripted:

Action:

Action:

Action:

Action:

Formula:

Action:

Formula:

JavaScript & HTML Code:

[None]

LotusScript Code:

(Globals)

Option Public
Option Declare
[None]

Name:

Alias:
Last Modification:
Comment:
Type:
Window Title Formula:

Include in Compose Menu:
Include in Query by Form:
Default Database Form:

SaveOptions

Text
[Not Assigned]
Computed
No
Disabled
No
"0"
No

\$Info

Text
[Not Assigned]
Computed for display
No
Disabled
No
ob := @If(ObservationsDisp = "";
""; @Word(ObservationsDisp;
""; 1));@If(ob = ""; "O0"; "O" +
ob);
No

Debug

Rich Text
[Not Assigned]
Editable
No
Disabled
No
No

Spacer

FormName

Spacer

Close

@Command([CloseWindow]);

Print

@Command([FilePrint]);

fmObservations

Observations
01/11/2023 03:22:01 PM
[Not Assigned]
Document
"HCL Domino License Analysis -
Results From " +
@Text(@Created; "D0T1Z0");
No
No
No

Automatically Refresh Fields:	No
Mail New Documents When Saving:	No
Store Form In Documents:	No
Inherit Default Field Values:	No
Updates Become Responses:	No
Retain Prior Versions As Responses:	No
Activate Objects When Composing:	No
Activate Objects When Editing:	No
Activate Objects When Reading:	No
Document Encryption Keys:	[None Assigned]
Composed Documents May Be Read By:	All Users
Form May Be Composed By:	All Users

Subcomponents:

Field:

Datatype:
Help Description:
Field Type:
Sign When Mailing/In Section:
Encryption:
Update Requires Editor Access:
Formula:
Is Scripted:

Field:

Datatype:
Help Description:
Field Type:
Sign When Mailing/In Section:
Encryption:
Update Requires Editor Access:
Formula:

Is Scripted:

Field:

Datatype:
Help Description:
Field Type:
Keyword User Interface:
Allow Values Not In List:
Allowable Keywords Formula:
Sign When Mailing/In Section:
Encryption:
Update Requires Editor Access:
Default Value Formula:
Is Scripted:

Field:

Datatype:
Help Description:
Field Type:
Sign When Mailing/In Section:
Encryption:
Update Requires Editor Access:
Formula:

Is Scripted:

SaveOptions

Text
[Not Assigned]
Computed
No
Disabled
No
"0"
No

\$Info

Text
[Not Assigned]
Computed for display
No
Disabled
No
ob := @If(ObservationsDisp = "";
""; @Word(ObservationsDisp;
"~"; 1));@If(ob = "";"O0"; "O" +
ob);
No

ObservationsDisp

Keywords
[Not Assigned]
Editable
Standard
No
Observations
No
Disabled
No
""

Observation

Text
[Not Assigned]
Computed for display
No
Disabled
No
val := ObservationsDisp;@If(val
= "";"", @Word(val; "~"; 2));
No

Hotspot

Type:
Caption:

Associated Data:

Action:

Action:

Action:

Action:

Formula:

Action:

Formula:

JavaScript & HTML Code:

[None]

LotusScript Code:**(Globals)**

```
Option Public
Option Declare
Dim openWS As NotesUIWorkspace
Dim openUI As NotesUIDocument
Dim openNote As NotesDocument
Dim thisUI As NotesUIDocument
Dim thisNote As NotesDocument
Dim openSession As NotesSession
Dim openDB As NotesDatabase
Dim openView As NotesView
```

(Form) User

```
Sub Postopen(Source As Notesuidocument)
    Set openWS=New NotesUIWorkspace
    Set openUI=source
    Set thisUI=Source
    Set openNote=source.Document
    Set thisNote=source.Document
    Set openSession=New NotesSession
    Set openDB=openSession.CurrentDatabase
End Sub
[None]
```

Name:

Last Modification:
Comment:
Type:
Window Title Formula:
Include in Compose Menu:
Include in Query by Form:
Default Database Form:
Automatically Refresh Fields:
Mail New Documents When Saving:
Store Form In Documents:
Inherit Default Field Values:
Updates Become Responses:
Retain Prior Versions As Responses:
Activate Objects When Composing:

Link

Click here for full instructions on enabling License Tracking f

https://help.hcltechsw.com/domin/12.0.0/admin/conf_licensetracking_t.html

Spacer

FormName

Spacer

Close

@Command([CloseWindow]);

Print

@Command([FilePrint]);

fmResults

01/11/2023 02:55:13 PM
[Not Assigned]
Document
@DbTitle
No
No
No
No
No
No
No
No
No
No
No

Activate Objects When Editing:
Activate Objects When Reading:
Document Encryption Keys:
Composed Documents May Be Read By:
Form May Be Composed By:

No
No
[None Assigned]
All Users
All Users

Subcomponents:

Field:

Datatype:
Help Description:
Field Type:
Sign When Mailing/In Section:
Encryption:
Update Requires Editor Access:
Formula:
Is Scripted:

Field:

Datatype:
Number Format:
Percentage (value * 100)%:
Parentheses on Negative Numbers:
Punctuated at Thousands:
Help Description:
Field Type:
Sign When Mailing/In Section:
Encryption:
Update Requires Editor Access:
Formula:

Is Scripted:

Field:

Datatype:
Number Format:
Percentage (value * 100)%:
Parentheses on Negative Numbers:
Punctuated at Thousands:
Help Description:
Field Type:
Sign When Mailing/In Section:
Encryption:
Update Requires Editor Access:
Formula:

Is Scripted:

Field:

Datatype:
Number Format:
Percentage (value * 100)%:
Parentheses on Negative Numbers:
Punctuated at Thousands:
Help Description:
Field Type:
Sign When Mailing/In Section:

SaveOptions

Text
[Not Assigned]
Computed when composed
No
Disabled
No
"0"
No

PersonDocCount

Number
General
No
No
No
[Not Assigned]
Computed
No
Disabled
No
val := @DbLookup("":
"NoCache"; "", "config"; "config";
@ThisName);@If(@IsError(val);
-20; @IsNumber(val); val; -1);
No

GroupDocCount

Number
General
No
No
No
[Not Assigned]
Computed
No
Disabled
No
val := @DbLookup("":
"NoCache"; "", "config"; "config";
@ThisName);@If(@IsError(val);
-20; @IsNumber(val); val; -1);
No

DAUserCount

Number
General
No
No
No
[Not Assigned]
Computed
No

Encryption:	Disabled
Update Requires Editor Access:	No
Formula:	val := @DbLookup("" : "NoCache"; "" ; "config"; "config"; @ThisName);@If(@IsError(val); -20; @IsNumber(val); val; -1);
Is Scripted:	No
Field:	LicenseTrackingDocCount
Datatype:	Number
Number Format:	General
Percentage (value * 100)%:	No
Parentheses on Negative Numbers:	No
Punctuated at Thousands:	No
Help Description:	[Not Assigned]
Field Type:	Computed
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Formula:	val := @DbLookup("" : "NoCache"; "" ; "config"; "config"; @ThisName);@If(@IsError(val); -20; @IsNumber(val); val; -1);
Is Scripted:	No
Field:	ValidUserCount
Datatype:	Number
Number Format:	General
Percentage (value * 100)%:	No
Parentheses on Negative Numbers:	No
Punctuated at Thousands:	No
Help Description:	[Not Assigned]
Field Type:	Computed
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Formula:	val := @DbLookup("" : "NoCache"; "" ; "config"; "config"; @ThisName);@If(@IsError(val); -10101; val);
Is Scripted:	No
Field:	MailUserCount
Datatype:	Number
Number Format:	General
Percentage (value * 100)%:	No
Parentheses on Negative Numbers:	No
Punctuated at Thousands:	No
Help Description:	[Not Assigned]
Field Type:	Computed
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Formula:	val := @DbLookup("" : "NoCache"; "" ; "config"; "config"; @ThisName);@If(@IsError(val); -10101; val);
Is Scripted:	No
Field:	EntitlementTrackingServers

Datatype:	Text
Input Multi-Value Separator(s):	Comma, Semicolon, New Line
Display Multi-Value Separator:	New Line
Help Description:	[Not Assigned]
Field Type:	Computed when composed
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Formula:	val := @DbLookup("": "NoCache"; ""; "config"; "config"; @ThisName);@If(@IsError(val); "< Error Retrieving Server Names >"; @Word(val; " "; 1));
Is Scripted:	No
Field:	UserReportLocation
Datatype:	Text
Help Description:	[Not Assigned]
Field Type:	Computed
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Formula:	val := @DbLookup("": "NoCache"; ""; "config"; "config"; @ThisName);@If(@IsError(val); "< Error Retrieving Server Names >"; @Word(val; " "; 1));
Is Scripted:	No
Field:	DAScanError
Datatype:	Text
Help Description:	[Not Assigned]
Field Type:	Computed
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Formula:	check := @DbLookup("": "NoCache"; ""; "config"; "config"; "DAScanError");@If(@IsError(check); ""; check = ""; "", "Note: there was an error scanning the additional Domino Directories");
Is Scripted:	No
Field:	LicenseTrackingServersMissing
Datatype:	Text
Input Multi-Value Separator(s):	Comma, Semicolon, New Line
Display Multi-Value Separator:	New Line
Help Description:	[Not Assigned]
Field Type:	Computed when composed
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Formula:	val := @DbLookup("": "NoCache"; ""; "config"; "config"; @ThisName);@If(@IsError(val);

Is Scripted:
Field:
Datatype:
Help Description:
Field Type:
Sign When Mailing/In Section:
Encryption:
Update Requires Editor Access:
Formula:

Is Scripted:
Hotspot
Type:
Caption:
Is Scripted:

Hotspot
Type:
Caption:
Formula:

Hotspot
Type:
Caption:
Formula:

Hotspot
Type:
Caption:
Formula:

Hotspot
Type:
Caption:
Formula:

"< Error Retrieving Server
Names >"; val);
No
LDAPDA
Text
[Not Assigned]
Computed when composed
No
Disabled
No
val := @DbLookup("" :
"NoCache"; "" ; "config"; "config";
@ThisName);@If(@IsError(val);
""; val);
No

Button
GetInfo
Yes, see below

Button
OpenReport
@SetTargetFrame("Main");
@Command([OpenFrameset];
"fsReport");

Action Hotspot
Person

@Do(@SetTargetFrame("Sideb
ar");
@Command([OpenFrameset];
"fsSidebar2"));
@Do(@SetTargetFrame("Main"
);
@PostedCommand([OpenView]
; "UserInfo"));

Action Hotspot
Documents

@Do(@SetTargetFrame("Sideb
ar");
@Command([OpenFrameset];
"fsSidebar2"));
@Do(@SetTargetFrame("Main"
);
@PostedCommand([OpenView]
; "UserInfo"));

Action Hotspot
Mail

@Do(@SetTargetFrame("Sideb
ar");
@Command([OpenFrameset];

Hotspot

Type:
Caption:
Formula:

```
"fsSidebar3"));
@Do(@SetTargetFrame("Main"
);
@PostedCommand([OpenView]
; "dagroups"));
```

Action Hotspot
Users

```
@Do(@SetTargetFrame("Sideb
ar");
@Command([OpenFrameset];
"fsSidebar3"));
@Do(@SetTargetFrame("Main"
);
@PostedCommand([OpenView]
; "dagroups"));
```

Hotspot

Type:
Caption:
Formula:

Action Hotspot
Deny

```
@Do(@SetTargetFrame("Sideb
ar");
@Command([OpenFrameset];
"fsSidebar3"));
@Do(@SetTargetFrame("Main"
);
@PostedCommand([OpenView]
; "dagroups"));
```

Hotspot

Type:
Caption:
Formula:

Action Hotspot
Access

```
@Do(@SetTargetFrame("Sideb
ar");
@Command([OpenFrameset];
"fsSidebar3"));
@Do(@SetTargetFrame("Main"
);
@PostedCommand([OpenView]
; "dagroups"));
```

Hotspot

Type:
Caption:
Formula:

Action Hotspot
Groups

```
@Do(@SetTargetFrame("Sideb
ar");
@Command([OpenFrameset];
"fsSidebar3"));
@Do(@SetTargetFrame("Main"
);
@PostedCommand([OpenView]
; "dagroups"));
```

Hotspot

Type:
Caption:

Action Hotspot
Deny

Formula:		<pre>@Do(@SetTargetFrame("Sidebar"); @Command([OpenFrameset]; "fsSidebar3")); @Do(@SetTargetFrame("Main"); @PostedCommand([OpenView] ; "dagroups"));</pre>
Hotspot		
Type:		Action Hotspot
Caption:		Access
Formula:		<pre>@Do(@SetTargetFrame("Sidebar"); @Command([OpenFrameset]; "fsSidebar3")); @Do(@SetTargetFrame("Main"); @PostedCommand([OpenView] ; "dagroups"));</pre>
Hotspot		
Type:		Action Hotspot
Caption:		Users
Formula:		<pre>@Do(@SetTargetFrame("Sidebar"); @Command([OpenFrameset]; "fsSidebar3")); @Do(@SetTargetFrame("Main"); @PostedCommand([OpenView] ; "dagroups"));</pre>
Hotspot		
Type:		Action Hotspot
Caption:		View Printable Report
Hotspot		
Type:		Action Hotspot
Caption:		Open Scan Log
Is Scripted:		Yes, see below
Field:		AdminServer
Datatype:		Text
Help Description:		[Not Assigned]
Field Type:		Computed when composed
Sign When Mailing/In Section:		No
Encryption:		Disabled
Update Requires Editor Access:		No
Formula:		<pre>val := @DbLookup("" : "NoCache"; "" ; "config"; "config"; "AdminServer"); @if(@IsError(val); "< Error Retrieving Admin Server Name >"; val);</pre>
Is Scripted:		No
Field:		ServerList
Datatype:		Keywords

Help Description:	[Not Assigned]
Field Type:	Editable
Keyword User Interface:	Standard
Allow Values Not In List:	No
Allowable Keywords Formula:	val := @DbLookup("" : "NoCache"; "" ; "config"; "config"; "ServerList");@If(@IsError(val); "< Error Retrieving Server Names >"; @Word(val; " "; 1) + " (" + @Word(val; " "; 2) + ")");
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Default Value Formula:	""
Is Scripted:	No
Field:	EntitlementTrackingServersDis
Datatype:	p
Help Description:	Keywords
Field Type:	[Not Assigned]
Keyword User Interface:	Editable
Allow Values Not In List:	Standard
Allowable Keywords Formula:	No
Sign When Mailing/In Section:	EntitlementTrackingServers
Encryption:	No
Update Requires Editor Access:	Disabled
Is Scripted:	No
Field:	DAInfo
Datatype:	Text
Help Description:	[Not Assigned]
Field Type:	Computed when composed
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Formula:	val1 := @DbLookup("" : "NoCache"; "" ; "config"; "config"; "DAName");val2 := @DbLookup("" : "NoCache"; "" ; "config"; "config"; "DAFilename"); @If(@IsError(val1); "< Error Retrieving Directory Assistance Information >"; val1 = ""; "Directory Assistance is not configured."; val1 + " (" + val2 + ")");
Is Scripted:	No
Field:	DAListExternal
Datatype:	Text
Help Description:	[Not Assigned]
Field Type:	Computed when composed
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Formula:	val := @DbLookup("" :

Is Scripted:	"NoCache"; "";
Hide When:	@ThisName);@If(@IsError(val);
Field:	"< Error Retrieving Server
Datatype:	Names >"; @Word(val; "~"; 1));
Input Multi-Value Separator(s):	No
Display Multi-Value Separator:	DAListExternal = "";
Help Description:	DAList
Field Type:	Text
Sign When Mailing/In Section:	Comma, Semicolon, New Line
Encryption:	New Line
Update Requires Editor Access:	[Not Assigned]
Formula:	Computed
	No
	Disabled
	No
	val := @DbLookup("" :
	"NoCache"; "";
	"DAList");ret :=
	@ReplaceSubstring(@If(@IsErr
	or(val); "< Error Retrieving
	Server Names >"; @Word(val;
	"~"; 1)); " "; "!!");
	@Trim(@If(DAListExternal = "";
	ret; @Replace(ret;
	DAListExternal; ""));
	No
	DAInfo = "";
	ObservationsDisp
	Keywords
	[Not Assigned]
	Editable
	Standard
	No
	val := @DbLookup("" :
	"NoCache"; "";
	"Observations");
	@If(@IsError(val); "";
	@Word(val; " "; 1));
	No
	Disabled
	No
	No
	sfmInfoOverlays
Subform:	
JavaScript & HTML Code:	
onClick	
onClick	
onClick	
onClick	
onClick	
Action Hotspot	
Client onClick	
	document.forms[0].GetInfo.click();
	document.forms[0].OpenReport.click()
onClick	
Field AdminServer	
onClick	

Field ServerList

onClick

onClick

onClick

Field DAList

onClick

LotusScript Code:**(Globals)**

Option Public

Option Declare

Use "slCore"

Dim thisUI As NotesUIDocument

Dim thisNote As NotesDocument

Sub SendInfo

On Error Goto errHandle

Dim msgStr As String

Dim userName As String

Dim company As String

Dim email As String

Dim phone As String

Dim hclNum As String

Dim perCount As Long

Dim daCount As Long

Dim validCount As Long

If db Is Nothing Then QO

If ws Is Nothing Then PO

Dim dialog As New NotesDocument(db)

If GetAdminValue ("Name")(0)="" Then

If Not ws.dialogbox("dlgUserInfo",t,t,f,f,f,f,"User Information",dialog,t,f,t) Then Exit Sub

If dialog.CompanyName(0)="" Then msgStr=msgStr+cr+tb+"- Company Name"

If dialog.HCLAccount(0)="" Then msgStr=msgStr+cr+tb+"- HCL Account #"

If dialog.Name(0)="" Then msgStr=msgStr+cr+tb+"- Your Name"

If dialog.Email(0)="" Then msgStr=msgStr+cr+tb+"- Your Email Address"

If dialog.Phone(0)="" Then msgStr=msgStr+cr+tb+"- Your Phone #"

If msgStr<>"" Then

Msgbox "Not all information was provided. All information is required."+cr+msgStr

Exit Sub

End If

SetAdminValue "Name",dialog.Name(0)

SetAdminValue "Email",dialog.Email(0)

SetAdminValue "Phone",dialog.Phone(0)

SetAdminValue "CompanyName",dialog.CompanyName(0)

SetAdminValue "HCLAccount",dialog.HCLAccount(0)

End If

SetAdminValue "PersonDocCount",thisNote.PersonDocCount(0)

SetAdminValue "DAUserCount",thisNote.DAUserCount(0)

SetAdminValue "ValidUserCount",thisNote.ValidUserCount(0)

Call ws.Editdocument(False,dialog,True,"blank",False,True)

' Call ws.settargetframe("Main")

' Call ws.OpenFrameset("fsReport")

%REM

userName=GetAdminValue("Name")(0)

company=GetAdminValue("CompanyName")(0)

phone=GetAdminValue("Phone")(0)

email=GetAdminValue("Email")(0)

hclNum=GetAdminValue("HCLAccount")(0)


```

perCount=GetAdminValue("PersonDocCount")(0)
daCount=GetAdminValue("DAUserCount")(0)
validCount=GetAdminValue("ValidUserCount")(0)
dialog.Name=userName
dialog.CompanyName=company
dialog.Phone=phone
dialog.Email=email
dialog.HCLAccount=hclNum
dialog.PersonDocCount=perCount
dialog.DAUserCount=daCount
dialog.ValidUserCount=validCount
If Not ws.dialogbox("dlgLicenseInfo",t,t,t,f,f,f,"User Information",dialog,t,t,f) Then Exit Sub
If session.NotesBuildVersion>449 Then
    Dim agent As NotesAgent
    Set agent=db.GetAgent("SendInfo")
    Call agent.Run()
Else
    dialog.Form="PrintReport"
    Dim newWS As New NotesUIWorkspace
    Call newWS.Editdocument(False,dialog,True,"blank",False,True)
End If
%ENDREM
errHandle:
    If Err=4840 Then Err=0
    If Err<>0 Then If Err<>4840 Then MsgBox("Error in function ' " & Getthreadinfo(1) & ": " & Getthreadinfo(1)
    Exit Sub
End Sub
(Form) fmWelcome
Sub Queryopen(Source As Notesuidocument, Mode As Integer, Isnewdoc As Variant, Continue As Variant)
'    QO
End Sub
Sub Postopen(Source As Notesuidocument)
'    PO
    Set thisUI=source
    Set thisNote=source.Document
End Sub
(Button) GetInfo
Sub Click(Source As Button)
    On Error Goto errHandle
    Dim msgStr As String
    If db Is Nothing Then QO
    If ws Is Nothing Then PO
    Dim dialog As New NotesDocument(db)
    If GetAdminValue ("Name")(0)="" Then
        If Not ws.dialogbox("dlgUserInfo",t,t,t,f,f,f,"User Information",dialog,t,f,t) Then Exit Sub
        If dialog.CompanyName(0)="" Then msgStr=msgStr+cr+tb+"- Company Name"
        If dialog.HCLAccount(0)="" Then msgStr=msgStr+cr+tb+"- HCL Account #"
        If dialog.Name(0)="" Then msgStr=msgStr+cr+tb+"- Your Name"
        If dialog.Email(0)="" Then msgStr=msgStr+cr+tb+"- Your Email Address"
        If dialog.Phone(0)="" Then msgStr=msgStr+cr+tb+"- Your Phone #"
        If msgStr<>"" Then
            MsgBox "Not all information was provided. All information is required."+cr+msgStr
            Exit Sub
        End If
        SetAdminValue "Name",dialog.Name(0)
        SetAdminValue "Email",dialog.Email(0)
    End If

```

```

        SetAdminValue "Phone",dialog.Phone(0)
        SetAdminValue "CompanyName",dialog.CompanyName(0)
        SetAdminValue "HCLAccount",dialog.HCLAccount(0)
    End If
    SetAdminValue "PersonDocCount",thisNote.PersonDocCount(0)
    SetAdminValue "DAUserCount",thisNote.DAUserCount(0)
    SetAdminValue "ValidUserCount",thisNote.ValidUserCount(0)
errHandle:
    If Err=4840 Then Err=0
    If Err<>0 Then If Err<>4840 Then MsgBox("Error in function ' " & Getthreadinfo(1) & ": " & Getthreadinfo(1)
    Exit Sub

```

(Hotspot) Open Scan Log

```

Sub Click(Source As Button)
    Dim logWS As New NotesUIWorkspace
    Dim logSession As New NotesSession
    Dim logDB As NotesDatabase
    Dim logView As NotesView
    Dim logNote As NotesDocument
    Set logDB=logSession.CurrentDatabase
    Set logView=logDB.GetView("(config)")
    Set logNote=logView.GetFirstDocument
    If logNote Is Nothing Then Exit Sub
    Dim logURL As String
    logURL=logNote.NotesURL
    Call logWS.URLOpen(logURL)

```

Name:

Last Modification:
 Comment:
 Type:
 Include in Compose Menu:
 Include in Query by Form:
 Default Database Form:
 Automatically Refresh Fields:
 Mail New Documents When Saving:
 Store Form In Documents:
 Inherit Default Field Values:
 Updates Become Responses:
 Retain Prior Versions As Responses:
 Activate Objects When Composing:
 Activate Objects When Editing:
 Activate Objects When Reading:
 Document Encryption Keys:
 Composed Documents May Be Read By:
 Form May Be Composed By:
 PostOpen:

fmSidebar

01/11/2023 02:55:14 PM
 [Not Assigned]
 Document
 No
 No
 No
 No
 No
 No
 No
 No
 No
 No
 No
 No
 [None Assigned]
 All Users
 All Users
 ENVIRONMENT Override := "";
 @True;

Subcomponents:

Field:

Datatype:
 Help Description:
 Field Type:
 Sign When Mailing/In Section:
 Encryption:
 Update Requires Editor Access:
 Formula:

SaveOptions

Text
 [Not Assigned]
 Computed when composed
 No
 Disabled
 No
 "0"

Is Scripted:	No
Field:	Option
Datatype:	Number
Number Format:	General
Percentage (value * 100)%:	No
Parentheses on Negative Numbers:	No
Punctuated at Thousands:	No
Help Description:	[Not Assigned]
Field Type:	Computed when composed
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Formula:	1
Is Scripted:	No
Subform:	sfmSidebarOptions
JavaScript & HTML Code:	
[None]	
LotusScript Code:	
[None]	
Name:	fmSidebar2
Last Modification:	01/11/2023 02:55:15 PM
Comment:	[Not Assigned]
Type:	Document
Include in Compose Menu:	No
Include in Query by Form:	No
Default Database Form:	No
Automatically Refresh Fields:	No
Mail New Documents When Saving:	No
Store Form In Documents:	No
Inherit Default Field Values:	No
Updates Become Responses:	No
Retain Prior Versions As Responses:	No
Activate Objects When Composing:	No
Activate Objects When Editing:	No
Activate Objects When Reading:	No
Document Encryption Keys:	[None Assigned]
Composed Documents May Be Read By:	All Users
Form May Be Composed By:	All Users
PostOpen:	ENVIRONMENT Override := ""; @True;
Subcomponents:	
Field:	SaveOptions
Datatype:	Text
Help Description:	[Not Assigned]
Field Type:	Computed when composed
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Formula:	"0"
Is Scripted:	No
Field:	Option
Datatype:	Number
Number Format:	General
Percentage (value * 100)%:	No
Parentheses on Negative Numbers:	No

Punctuated at Thousands:	No
Help Description:	[Not Assigned]
Field Type:	Computed when composed
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Formula:	2
Is Scripted:	No
Subform:	sfmSidebarOptions
JavaScript & HTML Code:	
[None]	
LotusScript Code:	
[None]	
Name:	fmSidebar3
Last Modification:	01/11/2023 02:55:15 PM
Comment:	[Not Assigned]
Type:	Document
Include in Compose Menu:	No
Include in Query by Form:	No
Default Database Form:	No
Automatically Refresh Fields:	No
Mail New Documents When Saving:	No
Store Form In Documents:	No
Inherit Default Field Values:	No
Updates Become Responses:	No
Retain Prior Versions As Responses:	No
Activate Objects When Composing:	No
Activate Objects When Editing:	No
Activate Objects When Reading:	No
Document Encryption Keys:	[None Assigned]
Composed Documents May Be Read By:	All Users
Form May Be Composed By:	All Users
PostOpen:	ENVIRONMENT Override := ""; @True;
Subcomponents:	
Field:	SaveOptions
Datatype:	Text
Help Description:	[Not Assigned]
Field Type:	Computed when composed
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Formula:	"0"
Is Scripted:	No
Field:	Option
Datatype:	Number
Number Format:	General
Percentage (value * 100)%:	No
Parentheses on Negative Numbers:	No
Punctuated at Thousands:	No
Help Description:	[Not Assigned]
Field Type:	Computed when composed
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No

Formula:	3
Is Scripted:	No
Subform:	sfmSidebarOptions
JavaScript & HTML Code:	
[None]	
LotusScript Code:	
[None]	
Name:	fmUser
Alias:	User
Last Modification:	01/11/2023 02:55:14 PM
Comment:	[Not Assigned]
Type:	Document
Window Title Formula:	"HCL Domino User Information"
Include in Compose Menu:	No
Include in Query by Form:	No
Default Database Form:	No
Automatically Refresh Fields:	No
Mail New Documents When Saving:	No
Store Form In Documents:	No
Inherit Default Field Values:	No
Updates Become Responses:	No
Retain Prior Versions As Responses:	No
Activate Objects When Composing:	No
Activate Objects When Editing:	No
Activate Objects When Reading:	No
Document Encryption Keys:	[None Assigned]
Composed Documents May Be Read By:	All Users
Form May Be Composed By:	All Users
Subcomponents:	
Field:	LicenseTimestamp
Datatype:	Time
Time Format:	01/25/2023
Help Description:	[Not Assigned]
Field Type:	Computed when composed
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Formula:	@ThisValue
Is Scripted:	No
Field:	NABNOTECERTEXPIRY
Datatype:	Time
Time Format:	01/25/2023
Help Description:	[Not Assigned]
Field Type:	Computed when composed
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Formula:	@ThisValue
Is Scripted:	No
Action:	Spacer
Action:	FormName
Action:	Spacer
Action:	Close
Formula:	@Command([CloseWindow]);
Action:	Print

Formula:
JavaScript & HTML Code:
[None]

@Command([FilePrint]);

LotusScript Code:

(Form) User

```
Sub Queryopen(Source As Notesuidocument, Mode As Integer, Isnewdoc As Variant, Continue As Variant)
    If source.EditMode Then
        Print "Document Edits are not allowed"
        Continue=False
    End If
End Sub
Sub Querymodechange(Source As Notesuidocument, Continue As Variant)
    If Not source.EditMode Then
        Print "Document Edits are not allowed"
        Continue=False
    End If
End Sub
[None]
```

Name:

Alias:
Last Modification:
Comment:
Type:
Window Title Formula:
Include in Compose Menu:
Include in Query by Form:
Default Database Form:
Automatically Refresh Fields:
Mail New Documents When Saving:
Store Form In Documents:
Inherit Default Field Values:
Updates Become Responses:
Retain Prior Versions As Responses:
Activate Objects When Composing:
Activate Objects When Editing:
Activate Objects When Reading:
Document Encryption Keys:
Composed Documents May Be Read By:
Form May Be Composed By:

(PrintReport)

PrintReport
01/11/2023 02:55:17 PM
[Not Assigned]
Document
"Your License Information"
No
No
No
No
No
No
No
No
No
No
No
No
[None Assigned]
All Users
All Users

Subcomponents:

Field:

Datatype:
Number Format:
Percentage (value * 100)%:
Parentheses on Negative Numbers:
Punctuated at Thousands:
Help Description:
Field Type:
Sign When Mailing/In Section:
Encryption:
Update Requires Editor Access:
Formula:

ValidUserCount

Number
General
No
No
No
[Not Assigned]
Computed when composed
No
Disabled
No
val := @DbLookup("" :
"NoCache"; "" ; "config"; "config";
@ThisName);@If(@IsError(val);
"< Error Retrieving Value >";

Is Scripted:
Field:
Datatype:
Number Format:
Percentage (value * 100)%:
Parentheses on Negative Numbers:
Punctuated at Thousands:
Help Description:
Field Type:
Sign When Mailing/In Section:
Encryption:
Update Requires Editor Access:
Formula:

val);
No
PersonDocCount
Number
General
No
No
No
[Not Assigned]
Computed when composed
No
Disabled
No
val := @DbLookup("" :
"NoCache"; "" ; "config"; "config";
@ThisName);@If(@IsError(val);
"< Error Retrieving Value >";
val);
No

Is Scripted:
Field:
Datatype:
Number Format:
Percentage (value * 100)%:
Parentheses on Negative Numbers:
Punctuated at Thousands:
Help Description:
Field Type:
Sign When Mailing/In Section:
Encryption:
Update Requires Editor Access:
Formula:

MailUserCount
Number
General
No
No
No
[Not Assigned]
Computed when composed
No
Disabled
No
val := @DbLookup("" :
"NoCache"; "" ; "config"; "config";
@ThisName);@If(@IsError(val);
"< Error Retrieving Value >";
val);
No

Is Scripted:
Field:
Datatype:
Number Format:
Percentage (value * 100)%:
Parentheses on Negative Numbers:
Punctuated at Thousands:
Help Description:
Field Type:
Sign When Mailing/In Section:
Encryption:
Update Requires Editor Access:
Formula:

DAUserCount
Number
General
No
No
No
[Not Assigned]
Computed when composed
No
Disabled
No
val := @DbLookup("" :
"NoCache"; "" ; "config"; "config";
@ThisName);@If(@IsError(val);
"< Error Retrieving Value >";
val);
No

Is Scripted:
Field:
Datatype:
Help Description:

Name
Text
[Not Assigned]

Field Type:	Computed when composed
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Formula:	val := @DbLookup("": "NoCache"; "", "config"; "config"; @ThisName);@If(@IsError(val); "< Error Retrieving Value >"; val);
Is Scripted:	No
Field:	Email
Datatype:	Text
Help Description:	[Not Assigned]
Field Type:	Computed when composed
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Formula:	val := @DbLookup("": "NoCache"; "", "config"; "config"; @ThisName);@If(@IsError(val); "< Error Retrieving Value >"; val);
Is Scripted:	No
Field:	Phone
Datatype:	Text
Help Description:	[Not Assigned]
Field Type:	Computed when composed
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Formula:	val := @DbLookup("": "NoCache"; "", "config"; "config"; @ThisName);@If(@IsError(val); "< Error Retrieving Value >"; val);
Is Scripted:	No
Field:	CompanyName
Datatype:	Text
Help Description:	[Not Assigned]
Field Type:	Computed when composed
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Formula:	val := @DbLookup("": "NoCache"; "", "config"; "config"; @ThisName);@If(@IsError(val); "< Error Retrieving Value >"; val);
Is Scripted:	No
Field:	HCLAccount
Datatype:	Text
Help Description:	[Not Assigned]
Field Type:	Computed when composed
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No

Formula:	val := @DbLookup("" : "NoCache"; "" ; "config"; "config"; @ThisName);@If(@IsError(val); "< Error Retrieving Value >"; val); No
Is Scripted:	
Hotspot	Action Hotspot
Type:	Show Scan Results
Caption:	@SetTargetFrame("Inner");
Formula:	@Command([OpenFrameset]; "fsInner");
Hotspot	Action Hotspot
Type:	Print This Report
Caption:	@Command([FilePrint]);
Formula:	
JavaScript & HTML Code:	
[None]	
LotusScript Code:	
[None]	

View Information

Name:	Deny Access Groups
Alias:	dagroups
Last Modification:	01/11/2023 02:55:14 PM
Comment:	[Not Assigned]
Default Database View:	No
No Response Hierarchy:	Disabled
Categories Initially:	Expanded
Unread Marks:	None
Refresh Frequency:	Automatic
Discard Index:	Never
View May Be Used By:	All Users
Time Relative Formulas:	No
Selection Formula:	SELECT Form = "Group";
Subcomponents:	
Column Number:	1
Column Title:	Lookup
Column Formula:	ListName
Column Width:	20 Characters
List Separator:	
Hidden:	Yes
Responses Only:	No
Icon:	No
Number Format:	General
Percentage (value * 100)%:	No
Parentheses on Negative Numbers:	No
Punctuated at Thousands:	No
Time Format:	01/25/2023 11:01:38 AM
Sorting:	Ascending
Justification:	Left
Totals:	None
Hide Detail Values:	No

Column Number:	2
Column Title:	Group Name
Column Formula:	@Name([Abbreviate]; ListName); 20 Characters
Column Width:	
List Separator:	
Hidden:	No
Responses Only:	No
Icon:	No
Number Format:	General
Percentage (value * 100)%:	No
Parentheses on Negative Numbers:	No
Punctuated at Thousands:	No
Time Format:	01/25/2023 11:01:38 AM
Sorting:	Ascending
Justification:	Left
Totals:	None
Hide Detail Values:	No
Column Number:	3
Column Title:	Last Modified
Column Formula:	NABNoteUpdated
Column Width:	14 Characters
List Separator:	
Hidden:	No
Responses Only:	No
Icon:	No
Number Format:	General
Percentage (value * 100)%:	No
Parentheses on Negative Numbers:	No
Punctuated at Thousands:	No
Time Format:	01/25/2023 11:01:38 AM
Sorting:	Ascending
Justification:	Left
Totals:	None
Hide Detail Values:	No
Column Number:	4
Column Title:	# Members
Column Formula:	@If(@IsAvailable(MemberQuan tity); MemberQuantity; 0); 8 Characters
Column Width:	
List Separator:	
Hidden:	No
Responses Only:	No
Icon:	No
Number Format:	Fixed (0 Decimal Places)
Percentage (value * 100)%:	No
Parentheses on Negative Numbers:	No
Punctuated at Thousands:	No
Time Format:	01/25/2023 11:01:38 AM
Sorting:	Ascending
Justification:	Right
Totals:	None
Hide Detail Values:	No
Column Number:	5
Column Title:	# SubGroups

Column Formula:	@If(@IsAvailable(SubgroupQuantity); SubgroupQuantity; 0); 8 Characters
Column Width:	
List Separator:	
Hidden:	No
Responses Only:	No
Icon:	No
Number Format:	Fixed (0 Decimal Places)
Percentage (value * 100)%:	No
Parentheses on Negative Numbers:	No
Punctuated at Thousands:	No
Time Format:	01/25/2023 11:01:38 AM
Sorting:	Ascending
Justification:	Right
Totals:	None
Hide Detail Values:	No
Column Number:	6
Column Title:	[Not Assigned]
Column Formula:	""
Column Width:	1 Characters
List Separator:	
Hidden:	No
Responses Only:	No
Icon:	No
Number Format:	General
Percentage (value * 100)%:	No
Parentheses on Negative Numbers:	No
Punctuated at Thousands:	No
Time Format:	01/25/2023 11:01:38 AM
Sorting:	No
Justification:	Left
Totals:	None
Hide Detail Values:	No
Column Number:	7
Column Title:	Description
Column Formula:	ListDescription
Column Width:	20 Characters
List Separator:	
Hidden:	No
Responses Only:	No
Icon:	No
Number Format:	General
Percentage (value * 100)%:	No
Parentheses on Negative Numbers:	No
Punctuated at Thousands:	No
Time Format:	01/25/2023 11:01:38 AM
Sorting:	Ascending
Justification:	Left
Totals:	None
Hide Detail Values:	No
JavaScript & HTML Code:	
[None]	
LotusScript Code:	
[None]	

Name:

Alias:
Last Modification:
Comment:
Default Database View:
No Response Hierarchy:
Categories Initially:
Unread Marks:
Refresh Frequency:
Discard Index:
View May Be Used By:
Time Relative Formulas:
Selection Formula:

Form Formula:

Subcomponents:**Column Number:**

Column Title:
Column Formula:

Column Width:
List Separator:
Hidden:
Responses Only:
Icon:
Number Format:
Percentage (value * 100)%:
Parentheses on Negative Numbers:
Punctuated at Thousands:
Time Format:
Sorting:
Justification:
Totals:
Hide Detail Values:

Column Number:

Column Title:
Column Formula:
Column Width:
List Separator:
Hidden:
Responses Only:
Icon:
Number Format:
Percentage (value * 100)%:
Parentheses on Negative Numbers:
Punctuated at Thousands:
Time Format:
Sorting:
Justification:
Totals:
Hide Detail Values:

Observations

Observations
01/11/2023 02:55:18 PM
[Not Assigned]
No
Disabled
Expanded
None
Automatic
Never
All Users
No
SELECT (Form = "History" |
Form = "config") &
@Text(ObservationsWarning) +
@Text(ObservationsSecurity) +
@Text(ObservationsInformation
) != "";
"Observations"

1

[Not Assigned]
@If(Form = "config"; "Current";
"Previous") + " Results";
4 Characters

No
No
No
General
No
No
No
01/25/2023 11:01 AM
Ascending, Categorized
Left
None
No

2

Date of Scan
@Created
14 Characters

No
No
No
General
No
No
No
01/25/2023 11:01 AM
Descending
Left
None
No

Column Number:

Column Title:
Column Formula:
Column Width:
List Separator:
Hidden:
Responses Only:
Icon:
Number Format:
Percentage (value * 100)%:
Parentheses on Negative Numbers:
Punctuated at Thousands:
Time Format:
Sorting:
Justification:
Totals:
Hide Detail Values:

3

Administration Server
AdminServer
20 Characters

No
No
No
General
No
No
No
01/25/2023 11:01:38 AM
Ascending
Left
None
No

Column Number:

Column Title:
Column Formula:

4

Observations
warn :=
@If(ObservationsWarning = "";
0; ObservationsWarning);sec :=
@If(ObservationsSecurity = "";
0; ObservationsSecurity);info :=
@If(ObservationsInformation =
""; 0; ObservationsInformation);
warn + sec + info;
9 Characters

Column Width:
List Separator:
Hidden:
Responses Only:
Icon:
Number Format:
Percentage (value * 100)%:
Parentheses on Negative Numbers:
Punctuated at Thousands:
Time Format:
Sorting:
Justification:
Totals:
Hide Detail Values:

No
No
No
Fixed (0 Decimal Places)
No
No
No
01/25/2023 11:01:38 AM
Ascending
Right
None
No

Column Number:

Column Title:
Column Formula:
Column Width:
List Separator:
Hidden:
Responses Only:
Icon:
Number Format:
Percentage (value * 100)%:
Parentheses on Negative Numbers:
Punctuated at Thousands:
Time Format:
Sorting:
Justification:

5

Warnings
ObservationsWarning
9 Characters

No
No
No
General
No
No
No
01/25/2023 11:01:38 AM
Ascending
Right

Totals:	None
Hide Detail Values:	No
Column Number:	6
Column Title:	# Security
Column Formula:	ObservationsSecurity
Column Width:	9 Characters
List Separator:	
Hidden:	No
Responses Only:	No
Icon:	No
Number Format:	General
Percentage (value * 100)%:	No
Parentheses on Negative Numbers:	No
Punctuated at Thousands:	No
Time Format:	01/25/2023 11:01:38 AM
Sorting:	Ascending
Justification:	Right
Totals:	None
Hide Detail Values:	No
Column Number:	7
Column Title:	# Info Only
Column Formula:	ObservationsInformation
Column Width:	9 Characters
List Separator:	
Hidden:	No
Responses Only:	No
Icon:	No
Number Format:	General
Percentage (value * 100)%:	No
Parentheses on Negative Numbers:	No
Punctuated at Thousands:	No
Time Format:	01/25/2023 11:01:38 AM
Sorting:	Ascending
Justification:	Right
Totals:	None
Hide Detail Values:	No
JavaScript & HTML Code:	
[None]	
LotusScript Code:	
[None]	
Name:	Scan History
Alias:	History
Last Modification:	01/11/2023 02:55:14 PM
Comment:	[Not Assigned]
Default Database View:	No
No Response Hierarchy:	Disabled
Categories Initially:	Expanded
Unread Marks:	None
Refresh Frequency:	Automatic
Discard Index:	Never
View May Be Used By:	All Users
Time Relative Formulas:	No
Selection Formula:	SELECT Form = "History";

Subcomponents:

Column Number:

Column Title:	1
Column Formula:	Date of Scan
Column Width:	@Created
List Separator:	14 Characters
Hidden:	No
Responses Only:	No
Icon:	No
Number Format:	General
Percentage (value * 100)%:	No
Parentheses on Negative Numbers:	No
Punctuated at Thousands:	No
Time Format:	01/25/2023 11:01 AM
Sorting:	Descending
Justification:	Left
Totals:	None
Hide Detail Values:	No

Column Number:

Column Title:	2
Column Formula:	Administration Server
Column Width:	AdminServer
List Separator:	20 Characters
Hidden:	No
Responses Only:	No
Icon:	No
Number Format:	General
Percentage (value * 100)%:	No
Parentheses on Negative Numbers:	No
Punctuated at Thousands:	No
Time Format:	01/25/2023 11:01:38 AM
Sorting:	Ascending
Justification:	Left
Totals:	None
Hide Detail Values:	No

Column Number:

Column Title:	3
Column Formula:	Auth. Users
Column Width:	ValidUserCount
List Separator:	9 Characters
Hidden:	No
Responses Only:	No
Icon:	No
Number Format:	General
Percentage (value * 100)%:	No
Parentheses on Negative Numbers:	No
Punctuated at Thousands:	No
Time Format:	01/25/2023 11:01:38 AM
Sorting:	Ascending
Justification:	Right
Totals:	None
Hide Detail Values:	No

Column Number:

Column Title:	4
Column Formula:	Person Docs
Column Width:	PersonDocCount
	9 Characters

List Separator:	
Hidden:	No
Responses Only:	No
Icon:	No
Number Format:	General
Percentage (value * 100)%:	No
Parentheses on Negative Numbers:	No
Punctuated at Thousands:	No
Time Format:	01/25/2023 11:01:38 AM
Sorting:	Ascending
Justification:	Right
Totals:	None
Hide Detail Values:	No
Column Number:	5
Column Title:	DA Groups
Column Formula:	GroupDocCount
Column Width:	9 Characters
List Separator:	
Hidden:	No
Responses Only:	No
Icon:	No
Number Format:	General
Percentage (value * 100)%:	No
Parentheses on Negative Numbers:	No
Punctuated at Thousands:	No
Time Format:	01/25/2023 11:01:38 AM
Sorting:	Ascending
Justification:	Right
Totals:	None
Hide Detail Values:	No
Column Number:	6
Column Title:	DA Users
Column Formula:	DAUserCount
Column Width:	9 Characters
List Separator:	
Hidden:	No
Responses Only:	No
Icon:	No
Number Format:	General
Percentage (value * 100)%:	No
Parentheses on Negative Numbers:	No
Punctuated at Thousands:	No
Time Format:	01/25/2023 11:01:38 AM
Sorting:	Ascending
Justification:	Right
Totals:	None
Hide Detail Values:	No
JavaScript & HTML Code:	
[None]	
LotusScript Code:	
[None]	
Name:	User Information
Alias:	UserInfo
Last Modification:	01/11/2023 02:55:12 PM

Comment:
Default Database View:
No Response Hierarchy:
Categories Initially:
Unread Marks:
Refresh Frequency:
Discard Index:
View May Be Used By:
Time Relative Formulas:
Selection Formula:

[Not Assigned]
Yes
Disabled
Collapsed
None
Automatic
Never
All Users
No
SELECT Form = "User";

Subcomponents:

Column Number:
Column Title:
Column Formula:

1
Lookup (hidden)
certExpired :=
CertificateExpired;httpPWSet :=
HTTPPasswordSet;da :=
DAUser;@If(UserType =
"Unknown / Unlicensed";
UserType; HTTPSecure =
"false" & httpPWSet = "true";
"Authorized"; httpPWSet = "true"
& da != "" & HTTPSecure =
"true" & certExpired = "true";
"Invalid"; da != "" ; "Invalid";
httpPWSet = "true";
"Authorized"; certExpired =
"true"; "Invalid"; "Authorized") + "
Users";
10 Characters

Column Width:
List Separator:
Hidden:
Responses Only:
Icon:
Number Format:
Percentage (value * 100)%:
Parentheses on Negative Numbers:
Punctuated at Thousands:
Time Format:
Sorting:
Justification:
Totals:
Hide Detail Values:

Yes
No
No
General
No
No
No
01/25/2023 11:01:38 AM
Ascending
Left
None
No

Column Number:

Column Title:
Column Formula:
Column Width:
List Separator:
Hidden:
Responses Only:
Icon:
Number Format:
Percentage (value * 100)%:
Parentheses on Negative Numbers:
Punctuated at Thousands:
Time Format:
Sorting:

2
[Not Assigned]
1
4 Characters

No
No
No
General
No
No
No
01/25/2023 11:01:38 AM
No

Justification:	Right
Totals:	Total
Hide Detail Values:	Yes
Column Number:	3
Column Title:	[Not Assigned]
Column Formula:	certExpired := CertificateExpired;httpPWSet := HTTPPasswordSet;da := DAUser;partial := @If(PartialDAUser = "", "", "\Partial Server Access"); category := @If(UserType = "Unknown / Unlicensed"; UserType; HTTPSecure = "false" & httpPWSet = "true"; "Authorized"; httpPWSet = "true" & da != "" & HTTPSecure = "true" & certExpired = "true"; "Invalid"; da != ""; "Invalid"; httpPWSet = "true"; "Authorized"; certExpired = "true"; "Invalid"; "Authorized") + " Users";@If(category = "Authorized Users"; category + @If(partial = "", "\Full Server Access"; partial); category); 2 Characters
Column Width:	No
List Separator:	No
Hidden:	No
Responses Only:	General
Icon:	No
Number Format:	No
Percentage (value * 100)%:	No
Parentheses on Negative Numbers:	No
Punctuated at Thousands:	No
Time Format:	01/25/2023 11:01:38 AM
Sorting:	Ascending, Categorized
Justification:	Left
Totals:	None
Hide Detail Values:	No
Column Number:	4
Column Title:	Full Name
Column Formula:	@Name([Abbreviate]; FullName); 20 Characters
Column Width:	No
List Separator:	No
Hidden:	No
Responses Only:	General
Icon:	No
Number Format:	No
Percentage (value * 100)%:	No
Parentheses on Negative Numbers:	No
Punctuated at Thousands:	No
Time Format:	01/25/2023 11:01:38 AM
Sorting:	Ascending
Justification:	Left

Totals:	None
Hide Detail Values:	No
Column Number:	5
Column Title:	[Not Assigned]
Column Formula:	@If(MailUser = ""; 0; 129);
Column Width:	2 Characters
List Separator:	
Hidden:	No
Responses Only:	No
Icon:	Yes
Number Format:	General
Percentage (value * 100)%:	No
Parentheses on Negative Numbers:	No
Punctuated at Thousands:	No
Time Format:	01/25/2023 11:01:38 AM
Sorting:	No
Justification:	Center
Totals:	None
Hide Detail Values:	No
Column Number:	6
Column Title:	User Type
Column Formula:	@If(@Contains(USERTYPE; "Unlicensed"); ""; USERTYPE);
Column Width:	10 Characters
List Separator:	
Hidden:	No
Responses Only:	No
Icon:	No
Number Format:	General
Percentage (value * 100)%:	No
Parentheses on Negative Numbers:	No
Punctuated at Thousands:	No
Time Format:	01/2023
Sorting:	Ascending
Justification:	Left
Totals:	None
Hide Detail Values:	No
Column Number:	7
Column Title:	License Type
Column Formula:	extNAB := ExternalNAB;extORG := ExternalORG;aclLevel := @Word(EntitlementList; "~"; 2); @If(@IsError(aclLevel); "< Unknown >"; (aclLevel = "Author" aclLevel = "Depositor" aclLevel = "Reader" aclLevel = "No Access") & (extNAB + extORG != ""); "CCX"; "CCB");
Column Width:	10 Characters
List Separator:	
Hidden:	No
Responses Only:	No
Icon:	No
Number Format:	General
Percentage (value * 100)%:	No
Parentheses on Negative Numbers:	No

Punctuated at Thousands:	No
Time Format:	01/2023
Sorting:	Ascending
Justification:	Left
Totals:	None
Hide Detail Values:	No
Column Number:	8
Column Title:	Highest ACL
Column Formula:	@Word(EntitlementList; "~"; 2);
Column Width:	10 Characters
List Separator:	
Hidden:	No
Responses Only:	No
Icon:	No
Number Format:	General
Percentage (value * 100)%:	No
Parentheses on Negative Numbers:	No
Punctuated at Thousands:	No
Time Format:	01/2023
Sorting:	Ascending
Justification:	Left
Totals:	None
Hide Detail Values:	No
Column Number:	9
Column Title:	Last Accessed
Column Formula:	LicenseTimestamp
Column Width:	10 Characters
List Separator:	
Hidden:	No
Responses Only:	No
Icon:	No
Number Format:	General
Percentage (value * 100)%:	No
Parentheses on Negative Numbers:	No
Punctuated at Thousands:	No
Time Format:	01/2023
Sorting:	Ascending
Justification:	Left
Totals:	None
Hide Detail Values:	No
Column Number:	10
Column Title:	Certificate Expiry
Column Formula:	NABNOTECERTEXPIRY
Column Width:	14 Characters
List Separator:	
Hidden:	No
Responses Only:	No
Icon:	No
Number Format:	General
Percentage (value * 100)%:	No
Parentheses on Negative Numbers:	No
Punctuated at Thousands:	No
Time Format:	01/25/2023
Sorting:	No
Justification:	Left
Totals:	None

Hide Detail Values:	No
Column Number:	11
Column Title:	Domino Directory
Column Formula:	NabName + " (" + NabServer + " " + NabPath + ")";
Column Width:	20 Characters
List Separator:	
Hidden:	No
Responses Only:	No
Icon:	No
Number Format:	General
Percentage (value * 100)%:	No
Parentheses on Negative Numbers:	No
Punctuated at Thousands:	No
Time Format:	01/25/2023 11:01:38 AM
Sorting:	Ascending
Justification:	Left
Totals:	None
Hide Detail Values:	No

JavaScript & HTML Code:

[None]

LotusScript Code:

(Action) Export

Option Declare

Sub Click(Source As Button)

Dim session As New NotesSession

Dim db As notesdatabase

Dim dc As NotesDocumentCollection

Dim view As NotesView

Dim ws As New notesuiworkspace

Dim options(3) As String

Dim selection As String

Dim folder As String

Set db=session.CurrentDatabase

options(0)="All User Documents"

options(1)="Authorized Users"

options(2)="Invalid Users"

options(3)="Unknown / Unlicensed Users"

selection=ws.Prompt(4,"What To Export","Which type of User Document do you want to export?","All User

If selection="" Then Exit Sub

folder=ws.Prompt(14, {}, {})

If selection=options(0) Then

Set view=db.GetView("\$All")

Set dc=view.GetAllDocumentsByKey("User")

Else

Set view=db.GetView("UserInfo")

Set dc=view.GetAllDocumentsByKey(selection)

End If

Export session, folder+"HCL License Analysis - User Export ("&selection&").csv", dc

Msgbox Cstr(dc.Count)+" User documents exported."&Chr(10)&Chr(10)+"Your CSV file is in the following l

End Sub

Sub Export(session As NotesSession,filePath As String, dc As NotesDocumentCollection)

On Error Goto LogErr

Dim ec As NotesViewEntryCollection

Dim ent As NotesViewEntry

Dim valString As String

```

Dim myStream As NotesStream
Dim fileNum As Integer, count As Integer, x As Integer
Dim fldArray(6) As String
Dim strVar As Variant
Dim doc As NotesDocument
Dim cs As String
cs=","
'Open the file
Set myStream = session.CreateStream()
Call myStream.Open(filePath, "ASCII")
Call myStream.Truncate()
'Write header row
valString="User Name,Email,Last Accessed,Certificate Expiry,In Deny Access Group,Domino Directory,No
Call myStream.Writetext(valString, 0)
'Write line for each entry
Set doc=dc.GetFirstDocument
While Not doc Is Nothing
    strVar=Evaluate(|@Name([Abbreviate];FullName)|,doc)
    valString=strVar(0)+cs+doc.InternetAddress(0)+cs
    strVar=Evaluate(|@Text(LicenseTimestamp;"D0T1Z0")|,doc)
    valString=valString+strVar(0)+cs
    strVar=Evaluate(|@Text(NABNOTECERTEXPIRY;"D0T1Z0")|,doc)
    valString=valString+strVar(0)+cs
    If doc.HasItem("DAUser") Then If doc.DAUser(0)="Yes" Then valString=valString+"Yes"
    valString=valString+cs
    valString=valString+doc.NABName(0)+(" "+doc.NABServer(0)+"|"+doc.NABPath(0)+")"+cs
    strVar=Evaluate(|@Implode(LicenseProtocol)|,doc)
    If Instr(1,strVar(0),"1",5)<>0 Then valString=valString+"Yes"
    valString=valString+cs
    If Instr(1,strVar(0),"2",5)<>0 Then valString=valString+"Yes"
    valString=valString+cs
    If Instr(1,strVar(0),"3",5)<>0 Then valString=valString+"Yes"
    valString=valString+cs
    If doc.HasItem("EntitlementList") Then
        strVar=Evaluate(|@Word(EntitlementList;"~";2)|,doc)
        valString=valString+strVar(0)+cs
        strVar=Evaluate(|@Word(EntitlementList;"~";1)|,doc)
        valString=valString+strVar(0)+cs
        strVar=Evaluate(|@Word(EntitlementList;"~";3)|,doc)
        valString=valString+strVar(0)+cs
        strVar=Evaluate(|@Word(EntitlementList;"~";4)|,doc)
        valString=valString+|"+strVar(0)
    Else
        valString=valString+cs+cs+cs
    End If
    Call myStream.Writetext(valString, 0)
    Set doc=dc.GetNextDocument(doc)
Wend
closeFile:
'Close the file
Call myStream.Close()
Exit Sub
LogErr:
Resume closeFile
Exit Sub
End Sub

```

Name:

Alias:
Last Modification:
Comment:
Default Database View:
No Response Hierarchy:
Categories Initially:
Unread Marks:
Refresh Frequency:
Discard Index:
View May Be Used By:
Time Relative Formulas:
Selection Formula:

(config)

config
01/11/2023 02:55:14 PM
[Not Assigned]
No
Disabled
Expanded
None
Automatic
Never
All Users
No
SELECT Form = "config";

Subcomponents:**Column Number:**

Column Title:
Column Formula:
Column Width:
List Separator:
Hidden:
Responses Only:
Icon:
Number Format:
Percentage (value * 100)%:
Parentheses on Negative Numbers:
Punctuated at Thousands:
Time Format:
Sorting:
Justification:
Totals:
Hide Detail Values:

1

Form
10 Characters

No
No
No
General
No
No
No
01/25/2023 11:01:38 AM
Ascending
Left
None
No

JavaScript & HTML Code:

[None]

LotusScript Code:

[None]

Name:

Alias:
Last Modification:
Comment:
Default Database View:
No Response Hierarchy:
Categories Initially:
Unread Marks:
Refresh Frequency:
Discard Index:
View May Be Used By:
Time Relative Formulas:
Selection Formula:

(dauser)

dauser
01/11/2023 02:55:16 PM
[Not Assigned]
No
Disabled
Expanded
None
Automatic
Never
All Users
No
SELECT Form = "User" &
DAUser != "" & UserType !=
"Unkown / Unlicensed";

Subcomponents:

Column Number:	1
Column Title:	#
Column Formula:	"1"
Column Width:	10 Characters
List Separator:	
Hidden:	No
Responses Only:	No
Icon:	No
Number Format:	General
Percentage (value * 100)%:	No
Parentheses on Negative Numbers:	No
Punctuated at Thousands:	No
Time Format:	01/25/2023 11:01:38 AM
Sorting:	Ascending
Justification:	Left
Totals:	None
Hide Detail Values:	No

JavaScript & HTML Code:

[None]

LotusScript Code:

[None]

Name:

Alias:	(groupmembers)
Last Modification:	members
Comment:	01/11/2023 02:55:18 PM
Default Database View:	[Not Assigned]
No Response Hierarchy:	No
Categories Initially:	Disabled
Unread Marks:	Expanded
Refresh Frequency:	None
Discard Index:	Automatic
View May Be Used By:	Never
Time Relative Formulas:	All Users
Selection Formula:	No
	SELECT Form = "Group" & Members != "";

Subcomponents:

Column Number:	1
Column Title:	Lookup
Column Formula:	@LowerCase(Members);
Column Width:	20 Characters
List Separator:	
Hidden:	No
Responses Only:	No
Icon:	No
Number Format:	General
Percentage (value * 100)%:	No
Parentheses on Negative Numbers:	No
Punctuated at Thousands:	No
Time Format:	01/25/2023 11:01:38 AM
Sorting:	Ascending
Justification:	Left
Totals:	None
Hide Detail Values:	No

Column Number:

Column Title:
Column Formula:
Column Width:
List Separator:
Hidden:
Responses Only:
Icon:
Number Format:
Percentage (value * 100)%:
Parentheses on Negative Numbers:
Punctuated at Thousands:
Time Format:
Sorting:
Justification:
Totals:
Hide Detail Values:

2

Server
@LowerCase(@Trim(Servers));
40 Characters

No
No
No
General
No
No
No
01/25/2023 11:01:38 AM
Ascending
Left
None
No

JavaScript & HTML Code:

[None]

LotusScript Code:

[None]

Name:

Alias:
Last Modification:
Comment:
Default Database View:
No Response Hierarchy:
Categories Initially:
Unread Marks:
Refresh Frequency:
Discard Index:
View May Be Used By:
Time Relative Formulas:
Selection Formula:

(invalid)

invalid
01/11/2023 02:55:16 PM
[Not Assigned]
No
Disabled
Expanded
None
Automatic
Never
All Users
No
certExpired :=
CertificateExpired;httpPWSet :=
HTTPPasswordSet;da :=
DAUser;check :=
@If(HTTPSecure = "false" &
httpPWSet = "true"; "Valid";
httpPWSet = "true" & da != "" &
HTTPSecure = "true" &
certExpired = "true"; "", da != "";
""; httpPWSet = "true"; "Valid";
certExpired = "true"; "", "Valid");
SELECT Form = "User" & check
= "" & UserType != "Unknown /
Unlicensed";

Subcomponents:**Column Number:**

Column Title:
Column Formula:
Column Width:
List Separator:

1

"all"
10 Characters

Hidden:	No
Responses Only:	No
Icon:	No
Number Format:	General
Percentage (value * 100)%:	No
Parentheses on Negative Numbers:	No
Punctuated at Thousands:	No
Time Format:	01/25/2023 11:01:38 AM
Sorting:	Ascending
Justification:	Left
Totals:	None
Hide Detail Values:	No
JavaScript & HTML Code:	
[None]	
LotusScript Code:	
[None]	
Name:	(mailuser)
Alias:	mailuser
Last Modification:	01/11/2023 02:55:19 PM
Comment:	[Not Assigned]
Default Database View:	No
No Response Hierarchy:	Disabled
Categories Initially:	Expanded
Unread Marks:	None
Refresh Frequency:	Automatic
Discard Index:	Never
View May Be Used By:	All Users
Time Relative Formulas:	No
Selection Formula:	certExpired := CertificateExpired;httpPWSet := HTTPPasswordSet;da := DAUser;check := @If(HTTPSecure = "false" & httpPWSet = "true"; "Valid"; httpPWSet = "true" & da != "" & HTTPSecure = "true" & certExpired = "true"; "" ; da != "" ; "" ; httpPWSet = "true"; "Valid"; certExpired = "true"; "" ; "Valid"); SELECT Form = "User" & check = "Valid" & UserType != "Unknown / Unlicensed" & MailUser != "";
Subcomponents:	
Column Number:	1
Column Title:	#
Column Formula:	"all"
Column Width:	10 Characters
List Separator:	
Hidden:	No
Responses Only:	No
Icon:	No
Number Format:	General

Percentage (value * 100)%:	No
Parentheses on Negative Numbers:	No
Punctuated at Thousands:	No
Time Format:	01/25/2023 11:01:38 AM
Sorting:	Ascending
Justification:	Left
Totals:	None
Hide Detail Values:	No
JavaScript & HTML Code:	
[None]	
LotusScript Code:	
[None]	

Name:	(parentgroups)
Alias:	parents
Last Modification:	01/11/2023 02:55:17 PM
Comment:	[Not Assigned]
Default Database View:	No
No Response Hierarchy:	Disabled
Categories Initially:	Expanded
Unread Marks:	None
Refresh Frequency:	Automatic
Discard Index:	Never
View May Be Used By:	All Users
Time Relative Formulas:	No
Selection Formula:	SELECT Form = "Group" & TopLevel != "";

Subcomponents:	
Column Number:	1
Column Title:	Lookup
Column Formula:	ListName
Column Width:	20 Characters
List Separator:	
Hidden:	No
Responses Only:	No
Icon:	No
Number Format:	General
Percentage (value * 100)%:	No
Parentheses on Negative Numbers:	No
Punctuated at Thousands:	No
Time Format:	01/25/2023 11:01:38 AM
Sorting:	Ascending
Justification:	Left
Totals:	None
Hide Detail Values:	No
JavaScript & HTML Code:	
[None]	
LotusScript Code:	
[None]	

Name:	(possibleusers)
Alias:	possibleusers
Last Modification:	01/11/2023 02:55:18 PM

Comment:
Default Database View:
No Response Hierarchy:
Categories Initially:
Unread Marks:
Refresh Frequency:
Discard Index:
View May Be Used By:
Time Relative Formulas:
Selection Formula:

[Not Assigned]
No
Disabled
Expanded
None
Automatic
Never
All Users
No
SELECT Form = "User" &
UserType != "Unknown /
Unlicensed";

Subcomponents:

Column Number:

Column Title:
Column Formula:
Column Width:
List Separator:
Hidden:
Responses Only:
Icon:
Number Format:
Percentage (value * 100)%:
Parentheses on Negative Numbers:
Punctuated at Thousands:
Time Format:
Sorting:
Justification:
Totals:
Hide Detail Values:

1
Lookup
FullName
20 Characters

No
No
No
General
No
No
No
01/25/2023 11:01:38 AM
Ascending
Left
None
No

Column Number:

Column Title:
Column Formula:
Column Width:
List Separator:
Hidden:
Responses Only:
Icon:
Number Format:
Percentage (value * 100)%:
Parentheses on Negative Numbers:
Punctuated at Thousands:
Time Format:
Sorting:
Justification:
Totals:
Hide Detail Values:

2
DA User
DAUSER
10 Characters

No
No
No
General
No
No
No
01/25/2023 11:01:38 AM
Ascending
Left
None
No

JavaScript & HTML Code:

[None]

LotusScript Code:

[None]

Name:

Alias:

(subgroupsunprocessed)
sgunprocessed

Last Modification:
Comment:
Default Database View:
No Response Hierarchy:
Categories Initially:
Unread Marks:
Refresh Frequency:
Discard Index:
View May Be Used By:
Time Relative Formulas:
Selection Formula:

01/11/2023 02:55:18 PM
[Not Assigned]
No
Disabled
Expanded
None
Automatic
Never
All Users
No
SELECT Form = "Group" &
TopLevel = "" & Servers != "" &
Processed = "";

Subcomponents:

Column Number:

Column Title:
Column Formula:
Column Width:
List Separator:
Hidden:
Responses Only:
Icon:
Number Format:
Percentage (value * 100)%:
Parentheses on Negative Numbers:
Punctuated at Thousands:
Time Format:
Sorting:
Justification:
Totals:
Hide Detail Values:

1
Lookup
ListName
20 Characters

No
No
No
General
No
No
No
01/25/2023 11:01:38 AM
Ascending
Left
None
No

JavaScript & HTML Code:

[None]

LotusScript Code:

[None]

Name:

Alias:
Last Modification:
Comment:
Default Database View:
No Response Hierarchy:
Categories Initially:
Unread Marks:
Refresh Frequency:
Discard Index:
View May Be Used By:
Time Relative Formulas:
Selection Formula:

(subgroups)

subgroups
01/11/2023 02:55:18 PM
[Not Assigned]
No
Disabled
Expanded
None
Automatic
Never
All Users
No
SELECT Form = "Group" &
TopLevel = "";

Subcomponents:

Column Number:

Column Title:
Column Formula:

1
Lookup
ListName

Column Width:	20 Characters
List Separator:	
Hidden:	No
Responses Only:	No
Icon:	No
Number Format:	General
Percentage (value * 100)%:	No
Parentheses on Negative Numbers:	No
Punctuated at Thousands:	No
Time Format:	01/25/2023 11:01:38 AM
Sorting:	Ascending
Justification:	Left
Totals:	None
Hide Detail Values:	No
JavaScript & HTML Code:	
[None]	
LotusScript Code:	
[None]	

Name:	(users)
Alias:	users
Last Modification:	01/11/2023 02:55:15 PM
Comment:	[Not Assigned]
Default Database View:	No
No Response Hierarchy:	Disabled
Categories Initially:	Expanded
Unread Marks:	None
Refresh Frequency:	Automatic
Discard Index:	Never
View May Be Used By:	All Users
Time Relative Formulas:	No
Selection Formula:	SELECT Form = "User";
Subcomponents:	
Column Number:	1
Column Title:	Lookup
Column Formula:	@LowerCase(FullName);
Column Width:	20 Characters
List Separator:	
Hidden:	No
Responses Only:	No
Icon:	No
Number Format:	General
Percentage (value * 100)%:	No
Parentheses on Negative Numbers:	No
Punctuated at Thousands:	No
Time Format:	01/25/2023 11:01:38 AM
Sorting:	Ascending
Justification:	Left
Totals:	None
Hide Detail Values:	No
Column Number:	2
Column Title:	DA User
Column Formula:	DAUSER
Column Width:	10 Characters

List Separator:	No
Hidden:	No
Responses Only:	No
Icon:	No
Number Format:	General
Percentage (value * 100)%:	No
Parentheses on Negative Numbers:	No
Punctuated at Thousands:	No
Time Format:	01/25/2023 11:01:38 AM
Sorting:	Ascending
Justification:	Left
Totals:	None
Hide Detail Values:	No
JavaScript & HTML Code:	
[None]	
LotusScript Code:	
[None]	
Name:	(valid)
Alias:	valid
Last Modification:	01/11/2023 02:55:15 PM
Comment:	[Not Assigned]
Default Database View:	No
No Response Hierarchy:	Disabled
Categories Initially:	Expanded
Unread Marks:	None
Refresh Frequency:	Automatic
Discard Index:	Never
View May Be Used By:	All Users
Time Relative Formulas:	No
Selection Formula:	certExpired := CertificateExpired;httpPWSet := HTTPPasswordSet;da := DAUser;check := @If(HTTPSecure = "false" & httpPWSet = "true"; "Valid"; httpPWSet = "true" & da != "" & HTTPSecure = "true" & certExpired = "true"; "" ; da != "" ; "" ; httpPWSet = "true"; "Valid"; certExpired = "true"; "" ; "Valid"); SELECT Form = "User" & check = "Valid" & UserType != "Unknown / Unlicensed";
Subcomponents:	
Column Number:	1
Column Title:	#
Column Formula:	"1"
Column Width:	10 Characters
List Separator:	
Hidden:	No
Responses Only:	No
Icon:	No
Number Format:	General

Percentage (value * 100)%:	No
Parentheses on Negative Numbers:	No
Punctuated at Thousands:	No
Time Format:	01/25/2023 11:01:38 AM
Sorting:	Ascending
Justification:	Left
Totals:	None
Hide Detail Values:	No
JavaScript & HTML Code:	
[None]	
LotusScript Code:	
[None]	
Name:	(\$All)
Last Modification:	01/11/2023 02:55:17 PM
Comment:	[Not Assigned]
Default Database View:	No
No Response Hierarchy:	Disabled
Categories Initially:	Expanded
Unread Marks:	None
Refresh Frequency:	Automatic
Discard Index:	Never
View May Be Used By:	All Users
Time Relative Formulas:	No
Selection Formula:	SELECT @All
Subcomponents:	
Column Number:	1
Column Title:	#
Column Formula:	1
Column Width:	10 Characters
List Separator:	
Hidden:	No
Responses Only:	No
Icon:	No
Number Format:	General
Percentage (value * 100)%:	No
Parentheses on Negative Numbers:	No
Punctuated at Thousands:	No
Time Format:	01/25/2023 11:01:38 AM
Sorting:	No
Justification:	Left
Totals:	Total
Hide Detail Values:	Yes
Column Number:	2
Column Title:	Form
Column Formula:	@If(Form = "config"; "1. Scan Log"; Form);
Column Width:	10 Characters
List Separator:	
Hidden:	No
Responses Only:	No
Icon:	No
Number Format:	General
Percentage (value * 100)%:	No

Parentheses on Negative Numbers:	No
Punctuated at Thousands:	No
Time Format:	01/25/2023 11:01:38 AM
Sorting:	Ascending, Categorized
Justification:	Left
Totals:	None
Hide Detail Values:	No
Column Number:	3
Column Title:	[Not Assigned]
Column Formula:	@Created
Column Width:	10 Characters
List Separator:	
Hidden:	No
Responses Only:	No
Icon:	No
Number Format:	General
Percentage (value * 100)%:	No
Parentheses on Negative Numbers:	No
Punctuated at Thousands:	No
Time Format:	01/25/2023 11:01:38 AM
Sorting:	No
Justification:	Left
Totals:	None
Hide Detail Values:	No
JavaScript & HTML Code:	
[None]	
LotusScript Code:	
[None]	

Subform Information

Name:	sfmInfoOverlays
Last Modification:	01/11/2023 02:55:17 PM
Include in Insert Subform list:	Yes
Include in New Form dialog:	No
Subcomponents:	
Field:	nl
Datatype:	Text
Help Description:	[Not Assigned]
Field Type:	Editable
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Default Value Formula:	@NewLine
Is Scripted:	No
Field:	AU
Datatype:	Text
Help Description:	[Not Assigned]
Field Type:	Editable
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Default Value Formula:	"This is the number of users who can access your HCL Domino server(s). Authorized

Users are calculated using the following logic:" + nl + "- There is a Person document for the user" + nl + "- They have a valid Notes Certificate" + nl + "- If no Notes Certificate exists, they have an Internet Password set for web access" + nl + "- The user's name is not listed in any Deny Access Groups that are in use on the HCL Domino server(s). If there are multiple HCL Domino servers, the user MUST be listed in the Deny Access Group used for each HCL Domino server to be considered an Invalid User" + nl + "- If the user is listed in a Deny Access Group that is in use, but the HCL Domino servers are incorrectly configured to still allow access using a web browser, they will be considered an Authorized User" + nl;

Is Scripted:
Field:
Datatype:
Help Description:
Field Type:
Sign When Mailing/In Section:
Encryption:
Update Requires Editor Access:
Default Value Formula:

No
PD
Text
[Not Assigned]
Editable
No
Disabled
No
"This is the total number of Person documents discovered in your HCL Domino server environment"

Is Scripted:
Field:
Datatype:
Help Description:
Field Type:
Sign When Mailing/In Section:
Encryption:
Update Requires Editor Access:
Default Value Formula:

No
DAG
Text
[Not Assigned]
Editable
No
Disabled
No
"This is the number of Deny Access Groups that are listed on the HCL Domino server document(s)."

Is Scripted:
Field:
Datatype:
Help Description:
Field Type:
Sign When Mailing/In Section:
Encryption:

No
DAU
Text
[Not Assigned]
Editable
No
Disabled

Update Requires Editor Access:
Default Value Formula:

No
"This is the total number of users in your HCL Domino environment that have been determined to have no access to the HCL Domino server(s). Deny Access Users are calculated using the following logic:" + nl + "- There is a Person document for the user" + nl + "- The Notes Certificate is expired" + nl + "- There is no Internet Password set" + nl + "- The user's name exists in a Deny Access Group that is in use on the HCL Domino server. If there are multiple HCL Domino servers, the user's name is listed in ALL Deny Access Groups in use for each HCL Domino server" + nl + "- The HCL Domino server(s) are configured to enforce security settings on all HTTP traffic";

Is Scripted:

Field:

Datatype:
Help Description:
Field Type:
Sign When Mailing/In Section:
Encryption:
Update Requires Editor Access:
Default Value Formula:

No
AS
Text
[Not Assigned]
Editable
No
Disabled
No

"This is the HCL Domino Administration server that was the first HCL Domino server in your environment. Most, if not all, server configuration and User details are stored on this HCL Domino server."

Is Scripted:

Field:

Datatype:
Help Description:
Field Type:
Sign When Mailing/In Section:
Encryption:
Update Requires Editor Access:
Default Value Formula:

No
DSE
Text
[Not Assigned]
Editable
No
Disabled
No

"This is the full list of HCL Domino servers in your environment. This list will also indicate if the HCL Domino server is Available or Offline. If the HCL Domino server is offline, this can indicate an issue with accessing the HCL Domino server or that the HCL Domino

Is Scripted:
Field:
Datatype:
Help Description:
Field Type:
Sign When Mailing/In Section:
Encryption:
Update Requires Editor Access:
Default Value Formula:

server is permanently offline, yet the Server Configuration document is still available."

No
ET
Text
[Not Assigned]
Editable
No
Disabled
No

"This is a list of the HCL Domino servers in your environment that have an Entitlement Tracking application scanning for user's implicit and explicit access to the various HCL Domino applications on the server."

Is Scripted:
Field:
Datatype:
Help Description:
Field Type:
Sign When Mailing/In Section:
Encryption:
Update Requires Editor Access:
Default Value Formula:

No
DA
Text
[Not Assigned]
Editable
No
Disabled
No

"If your HCL Domino servers use Directory Assistance, this value will indicate the Title and file location of the Directory Assistance Catalog application"

Is Scripted:
Field:
Datatype:
Help Description:
Field Type:
Sign When Mailing/In Section:
Encryption:
Update Requires Editor Access:
Default Value Formula:

No
ADD
Text
[Not Assigned]
Editable
No
Disabled
No

"This is a list of all additional HCL Domino Directories that are configured in the Directory Assistance Catalog. If one of these values are in blue, it is assumed that all external users (CCX Users) are managed in that specific Domino Directory."

Is Scripted:
Field:
Datatype:
Help Description:
Field Type:
Sign When Mailing/In Section:
Encryption:
Update Requires Editor Access:
Default Value Formula:

No
OB
Text
[Not Assigned]
Editable
No
Disabled
No

"This is a list of findings that

Is Scripted:
Field:
Datatype:
Help Description:
Field Type:
Sign When Mailing/In Section:
Encryption:
Update Requires Editor Access:
Default Value Formula:

Is Scripted:
Field:
Datatype:
Help Description:
Field Type:
Sign When Mailing/In Section:
Encryption:
Update Requires Editor Access:
Formula:

Is Scripted:
Hotspot
Type:
Caption:
Formula:

were observed during the environment scans. These will be in one of three categories:" + nl + nl + "- \"Informational\": items that do not need immediate attention but you should be aware of" + nl + "- \"Warnings\": items that you need to be aware of as they can cause performance issues in your environment" + nl + "- \"Security\": items that you should address immediately to ensure that your environment is fully secured";

No
MU
Text
[Not Assigned]
Editable
No
Disabled
No
"This is the total number of Mail Users discovered in your HCL Domino server environment"

No
TitleDisp
Text
[Not Assigned]
Computed for display
No
Disabled
No
info := ShowInfo;options := "AU" : "PD" : "DAG" : "DAU" : "AS" : "DSE" : "ET" : "DA" : "ADD" : "OB" : "MU";visual := "Authorized Users" : "Person Documents" : "Deny Access Groups" : "Deny Access Users" : "Administration Server" : "All Domino Servers in Environment" : "Entitlement Tracking Servers" : "Directory Assistance Information" : "Additional Domino Directories" : "Observations" : "Mail Users";@Replace(info; options; visual);
No

Action Hotspot
X
FIELD ShowInfo := "";
@Command([ToolsRefreshAllD ocs]);

Field:

Datatype:
Help Description:
Field Type:
Sign When Mailing/In Section:
Encryption:
Update Requires Editor Access:
Formula:

Is Scripted:

JavaScript & HTML Code:

[None]

LotusScript Code:

[None]

Name:

Last Modification:
Include in Insert Subform list:
Include in New Form dialog:

Subcomponents:

Graphic Button
Formula:

Hotspot

Type:
Caption:
Formula:

Graphic Button
Formula:

Hotspot

Type:
Caption:
Formula:

@Command([RefreshHideFormulas]);
InfoDisp
Text
[Not Assigned]
Computed for display
No
Disabled
No
info := ShowInfo;options := "AU" : "PD" : "DAG" : "DAU" : "AS" : "DSE" : "ET" : "DA" : "ADD" : "OB" : "MU";visual := AU : PD : DAG : DAU : AS : DSE : ET : DA : ADD : OB : MU;
@Replace(info; options; visual);
No

sfmSidebarOptions

01/11/2023 02:55:16 PM
Yes
No

FIELD Option := 1;
@Command([RefreshHideFormulas]);
@SetTargetFrame("Main");
@PostedCommand([OpenFrameset]; "fsResults");

Action Hotspot
Scan Results
FIELD Option := 1;
@Command([RefreshHideFormulas]);
@SetTargetFrame("Main");
@PostedCommand([OpenFrameset]; "fsResults");

FIELD Option := 2;
@Command([RefreshHideFormulas]);
@SetTargetFrame("Main");
@PostedCommand([OpenView]; "UserInfo");

Action Hotspot
User Information
FIELD Option := 2;
@Command([RefreshHideFormulas]);

		<pre>@SetTargetFrame("Main"); @PostedCommand([OpenView] ; "UserInfo");</pre>
Graphic Button	Formula:	
		<pre>FIELD Option := 3; @Command([RefreshHideForm ulas]); @SetTargetFrame("Main"); @PostedCommand([OpenView] ; "dagroups");</pre>
Hotspot	Type: Caption: Formula:	<p>Action Hotspot</p> <p>Deny Access Groups</p> <pre>FIELD Option := 3; @Command([RefreshHideForm ulas]); @SetTargetFrame("Main"); @PostedCommand([OpenView] ; "dagroups");</pre>
Graphic Button	Formula:	
		<pre>FIELD Option := 5; @Command([RefreshHideForm ulas]); @SetTargetFrame("Main"); @PostedCommand([OpenView] ; "Observations");</pre>
Hotspot	Type: Caption: Formula:	<p>Action Hotspot</p> <p>Observations</p> <pre>FIELD Option := 5; @Command([RefreshHideForm ulas]); @SetTargetFrame("Main"); @PostedCommand([OpenView] ; "Observations");</pre>
Graphic Button	Formula:	
		<pre>FIELD Option := 4; @Command([RefreshHideForm ulas]); @SetTargetFrame("Main"); @PostedCommand([OpenView] ; "History");</pre>
Hotspot	Type: Caption: Formula:	<p>Action Hotspot</p> <p>Previous Results</p> <pre>FIELD Option := 4; @Command([RefreshHideForm ulas]); @SetTargetFrame("Main"); @PostedCommand([OpenView] ; "History");</pre>
Graphic Button	Formula:	
		<pre>@SetTargetFrame("Inner"); @PostedCommand([OpenFram eset]; "fsLaunch");</pre>
Hotspot		

Type:
Caption:
Formula:

Action Hotspot
Return to Wizard
@SetTargetFrame("Inner");
@PostedCommand([OpenFrame
set]; "fsLaunch");

JavaScript & HTML Code:

[None]

LotusScript Code:

[None]

Code Library Information

Name:

Last Modification:

LotusScript Code:

```
Option Public
Option Declare
Use "slUtilities"
Dim dadb As NotesDatabase
Dim dirArray() As Variant
Dim svrName As NotesName
Dim nab As NotesDatabase
Dim svrNote As NotesDocument
Dim nabFilename As String
Const adminProfileName="environmentinfo"
Const userProfileName="userinfo"
' these are to allow more than 65536 Group names. With these variables we will be able to load up 1,000,000 G
Dim allGroupNames1 As Variant
Dim allGroupNames2 As Variant
Dim allGroupNames3 As Variant
Dim allGroupNames4 As Variant
Dim allGroupNames5 As Variant
Dim allGroupNames6 As Variant
Dim allGroupNames7 As Variant
Dim allGroupNames8 As Variant
Dim allGroupNames9 As Variant
Dim allGroupNames10 As Variant
Dim allGroupNames11 As Variant
Dim allGroupNames12 As Variant
Dim allGroupNames13 As Variant
Dim allGroupNames14 As Variant
Dim allGroupNames15 As Variant
Dim allGroupNames16 As Variant
Dim allGroupNames17 As Variant
Dim allGroupNames18 As Variant
Dim allGroupNames19 As Variant
Dim allGroupNames20 As Variant
Dim allGroupNames21 As Variant
Dim allGroupNames22 As Variant
Dim allGroupNames23 As Variant
Dim allGroupNames24 As Variant
Dim allGroupNames25 As Variant
Dim allGroupNames26 As Variant
Dim allGroupNames27 As Variant
Dim allGroupNames28 As Variant
```

slCore

01/11/2023 02:55:13 PM


```

Dim allGroupNames29 As Variant
Dim allGroupNames30 As Variant
Dim allGroupNames31 As Variant
Dim allGroupNames32 As Variant
Dim allGroupNames33 As Variant
Dim allGroupNames34 As Variant
Dim allGroupNames35 As Variant
Dim allGroupNames36 As Variant
Dim allGroupNames37 As Variant
Dim allGroupNames38 As Variant
Dim allGroupNames39 As Variant
Dim allGroupNames40 As Variant
Sub Initialize
    cr=Chr(10)
    tb=Chr(9)
    ReDim orgArray(0)
    nabFilename="names.nsf"
End Sub
Function CheckServerSettings() As Boolean
    If Not debug Then On Error GoTo errHandle
    LogStep norm,CStr(Now)+tb+|"CheckServerSettings" - begin|
    Dim nabView As NotesView
    Dim nabdoc As NotesDocument
    Dim globaldoc As NotesDocument
    Dim docFound As Boolean
    Dim licArray As Variant
    Dim missingArray As Variant
    Dim svrArray As Variant
    Dim splitVar As Variant
    Dim licDB As NotesDatabase
    Dim licView As NotesView
    Dim licNote As NotesDocument
    Dim licDBCreated As NotesDateTime
    Dim response As Integer
    Dim x As Integer,y As Integer,z As Integer
    ReDim licArray(0)
    licArray(0)=""
    ReDim missingArray(0)
    missingArray(0)=""
    ReDim svrArray(0)
    svrArray(0)=""
    Set nabView=nab.GetView("($Servers)")
    If nabView Is Nothing Then
        LogStep warn,CStr(Now)+tb+|In "CheckServerSettings" - View cannot be located - "($Servers)"|
        Set nabView=nab.GetView("($ServersLookup)")
    End If
    If nabView Is Nothing Then
        LogStep warn,CStr(Now)+tb+|In "CheckServerSettings" - View cannot be located - "($ServersLookup)"|
        Set nabView=nab.GetView("Servers")
    End If
    If nabView Is Nothing Then
        errString=|A View is missing form your Domino Directory - "Servers"|
        errNum=9999
        errLine=35
        errCritical=t
        Exit Function
    End If

```

```

End If
LogStep norm,CStr(Now)+tb+|In "CheckServerSettings" - locating Admin server doc|
Set nabdoc=nabView.GetDocumentByKey(svrName.Canonical,f)
If nabdoc Is Nothing Then
    Set nabdoc=nabView.GetFirstDocument
    While Not nabDoc Is Nothing
        If LCase(nabdoc.ServerName(0))=LCase(svrName.Canonical) Then GoTo nextStep
        Set nabdoc=nabView.GetNextDocument(nabdoc)
    Wend
End If
nextStep:
If nabdoc Is Nothing Then GoTo cannotGetValues
LogStep norm,CStr(Now)+tb+|In "CheckServerSettings" - Admin server doc located|
LogStep norm,CStr(Now)+tb+|In "CheckServerSettings" - checking for Directory Assistance info|
' check for Directory Assistance configuration. If it is enabled, get all additional Domino Directories
SetAdminValue "DAScanError",""
If nabdoc.Hasitem("MasterAddressBook") Then
    If nabdoc.MasterAddressBook(0)<>"" Then
        SetAdminValue "DAFilename",nabdoc.MasterAddressBook(0)
        Set dadb=session.Getdatabase(svrName.Canonical,nabdoc.MasterAddressBook(0))
        If Not dadb.Isopen Then Call dadb.Open("","")
        If dadb.Isopen Then
            SetAdminValue "DAName",dadb.Title
            If Not GetDADirectories Then SetAdminValue "DAScanError","true"
        End If
    End If
End If
' check for the Entitlement Tracking db on the accessible servers
LogStep norm,CStr(Now)+tb+|In "CheckServerSettings" - checking for Entitlement Tracking apps|
svrArray=GetAdminValue("ServerList")
ReDim licArray(0)
licArray(0)=""
y=0
If svrArray(0)<>"" Then
    For x=0 To UBound(svrArray)
        splitVar=Split(svrArray(x),"|")
        If splitVar(1)="Available" Then
            Set licDB=session.Getdatabase(splitVar(0),"entitlementtrack.ncf")
            If Not licDB.isopen Then Call licDB.Open("","")
            If licDB.isopen Then
                ReDim Preserve licArray(y)
                licArray(y)=splitVar(0)
                y=y+1
            End If
        End If
    Next
End If
SetAdminValue "EntitlementTrackingServers",licArray
' ask questions here
' Question 1: - User Report Tool
If licArray(0)="" Then
    LogStep norm,CStr(Now)+tb+|In "CheckServerSettings" - asking for User Report Tool output|
    ' there is no Entitlement Tracking dbs - ask for a location of a User Report Tool if they have one unless
    If GetAdminValue("UserReportLocation")(0)="" Then
        response=MsgBox("For better analysis of CCB / CCX licenses, we require the results from the User Report Tool")
        If response=6 Then

```

```

        SetAdminValue "PauseTimer","true"
        SelectURT
        SetAdminValue "PauseTimer",""
    End If
End If
End If
' Question 2: External Users
Dim ask As Boolean
Dim msgStr As String
If UBound(GetAdminValue("Orgs"))>0 Then ask=t
If GetAdminValue("DAList")(0)<>"" Then ask=t
If GetAdminValue("DAListExternal")(0)<>"" Then ask=f
If GetAdminValue("OrgsExternal")(0)<>"" Then ask=f
If ask Then
    LogStep norm,CStr(Now)+tb+|In "CheckServerSettings" - asking about external users|
    msgStr=|Does your organization have "External Users" configured in your environment?|+cr+cr+|An "E
    |who is assigned Authorized User Access To the Program with restricted maximum level of a HCL Domino
    |but may be reassigned To another user after thirty (30) days of inactivity.|
    response=MsgBox(msgStr,36,"External Users?")
    If response=6 Then
        Dim dialog As New NotesDocument(db)
        If ws Is Nothing Then Set ws=New NotesUIWorkspace
        SetAdminValue "PauseTimer","true"
        If ws.Dialogbox("dlgExternalQA", t, t, f, f, f, f, "", dialog, t, f, t) Then
            SetAdminValue "DAListExternal", dialog.DAListExternal(0)
            SetAdminValue "OrgsExternal", dialog.Getfirstitem("OrgsExternal").Values
        End If
        SetAdminValue "PauseTimer","true"
    End If
End If
' check for the License Tracking db on the accessible servers
LogStep norm,CStr(Now)+tb+|In "CheckServerSettings" - checking for License Tracking apps|
CheckLicenseTrackingSettings
exitThis:
    LogStep norm,CStr(Now)+tb+|"CheckServerSettings" - finished|
    CheckServerSettings=t
    Exit Function
cannotGetValues:
    MsgBox "Could not locate information about the HCL Domino server from the HCL Domino Directory. Please
    Exit Function
perdocError:
    MsgBox "Could not scan for Person documents in your HCL Domino Directory. Please ensure that you can
    Exit Function
errHandle:
    If Err<>0 Then
        errString=Error
        errNum=Err
        errLine=Erl
        ReportError "There was an Error in the (CheckServerSettings) function. The details of the Error are bel
    End If
    Exit Function
End Function
Function GetDenyAccessGroups() As Boolean
    If Not debug Then On Error GoTo errHandle
    %REM
        so here's what we need to do:

```

- Get the list of "Deny Access" groupd that were found on the server docs and put them into an array (r
- Verify that thr Group exists in the main NAB, if a Group does not exist then remove it form the list
- for each remaining Group name make a DA Group doc in this db
- Get the matching Group doc from the main NAB
- move all Members onto the DA Group doc in this db.
 - while scanning the Members, check to see if they are also a Group value. If they are, then make a
 - remove the value from the Members list and add the Group name to an additional array for furt
- Once all original DA Groups have been processed, start processing any additional Groups that were f

%END REM

LogStep norm,CStr(Now)+tb+|"GetDenyAccessGroups" - begin|

Dim allGroupNames() As String

Dim serverName As NotesName

Dim serverArray As Variant

Dim daGroups() As String

Dim foundGroups() As String

Dim missingGroups() As String

Dim daGroupServer() As String

Dim subGroups() As String

Dim memberNames() As String

Dim splitVar As Variant

Dim missingDA As Boolean

Dim daGroupView As NotesView

Dim groupView As NotesView

Dim daView As NotesView

Dim nabNote As NotesDocument

Dim daNote As NotesDocument

Dim item As NotesItem

Dim w As Long, x As Long, y As Long, z As Long

Dim savelt As Boolean

Dim neededFields(7) As String

neededFields(0)="listname"

neededFields(1)="nabnoteupdated"

neededFields(2)="memberquantity"

neededFields(3)="subgroupquantity"

neededFields(4)="listdescription"

neededFields(5)="servers"

neededFields(6)="parentgroup"

neededFields(7)="form"

Const low=-32768

ReDim daGroups(0)

ReDim foundGroups(0)

ReDim missingGroups(0)

ReDim daGroupServer(0)

ReDim subGroups(0)

ReDim memberNames(0)

Set daView=db.Getview("dagroups")

Call daView.Refresh()

daView.Autoupdate=f

' scan the list of servers to first make sure they are Available, then grab a list of the Deny Access Groups co

serverArray=GetAdminValue("ServerList")

If Not IsArray(serverArray) Then errNum=9003:errLine=10:errString=|List of HCL Domino Servers is not se

If serverArray(0)="" Then errNum=9003:errLine=10:errString=|List of HCL Domino Servers is not set|:GoTo

For x=0 To UBound(serverArray)

 If InStr(1,serverArray(x),"Available",5)<>0 Then

 splitVar=Split(serverArray(x),"|")

 Set serverName=New NotesName(splitVar(0))

```

splitVar=StrToken(serverArray(x),"|",3)
If splitVar="" Then      ' at least one of the Servers is not using Deny Access Groups. This means
    missingDA=t
    Observation 1005,|The HCL Domino server "|+serverName.Abbreviated+" is not using any De
    LogStep sec,CStr(Now)+tb+|The HCL Domino server "|+serverName.Abbreviated+" is not usi
ElseIf InStr(splitVar,"~") Then
    splitVar=Split(splitVar,"~")
    For y=0 To UBound(splitVar)
        If Not IsNumeric(ArrayGetIndex(daGroups,splitVar(y))) Then
            ReDim Preserve daGroups(z)
            daGroups(z)=splitVar(y)
            z=z+1
        End If
    Next
Else
    If Not IsNumeric(ArrayGetIndex(daGroups,splitVar)) Then
        ReDim Preserve daGroups(z)
        daGroups(z)=splitVar
        z=z+1
    End If
End If
End If
Next
' setup the Views from the Domino Directory
Set groupView=nab.GetView("($VIMGroups)" )
Call groupView.Refresh()
groupView.Autoupdate=f
Set daGroupView=nab.Getview("DenyLists")
Call daGroupView.Refresh()
daGroupView.Autoupdate=f
%rem
x=low
' load all available Group Names into an array for use later in this function
Set nabNote=groupView.Getfirstdocument()
While Not nabNote Is Nothing
    ReDim Preserve allGroupNames(low To x)
    allGroupNames(x)=LCase(nabNote.ListName(0))
    x=x+1
    Set nabNote=groupView.Getnextdocument(nabNote)
Wend
Set nabNote=daGroupView.Getfirstdocument()
While Not nabNote Is Nothing
    ReDim Preserve allGroupNames(low To x)
    allGroupNames(x)=LCase(nabNote.ListName(0))
    x=x+1
    Set nabNote=daGroupView.Getnextdocument(nabNote)
Wend
x=0
Stop
%endrem
GetAllGroupNames groupView,daGroupView
' now check to see if the Deny Access Group actually exists in the Domino Directory - if a DA Group is miss
y=0
Dim memberName As NotesName
Dim groupName As NotesName
Dim checkVal As String

```

```

Dim processedGroups() As String
Dim allSubGroups() As String
Dim firstScan As Boolean
ReDim processedGroups(0)
ReDim allSubGroups(0)
firstScan=t
startScan:
For x=0 To UBound(daGroups)
    If Not IsNumeric(ArrayGetIndex(processedGroups,LCase(daGroups(x)))) Then ' if the group name has
        Set groupName=New NotesName(daGroups(x))
        Set nabNote=daGroupView.Getdocumentbykey(groupName.Abbreviated,t)
        If nabNote Is Nothing Then Set nabNote=groupView.Getdocumentbykey(groupName.Abbreviated,
        If nabNote Is Nothing Then
            ReDim missingGroups(y)
            missingGroups(y)=daGroups(x)
            Observation 1006,|The Deny Access Group named "|+daGroups(x)+|" appears to be missing fr
            LogStep norm,CStr(Now)+tb+|The Deny Access Group named "|+daGroups(x)+|" appears to b
        Else
            ' add the Group name to the "processed" array
            z=UBound(processedGroups)
            If processedGroups(z)<>"" Then z=z+1
            ReDim Preserve processedGroups(z)
            processedGroups(z)=LCase(daGroups(x))
            Set daNote=daView.Getdocumentbykey(groupName.Abbreviated,t)
            If daNote Is Nothing Then
                Set daNote=db.Createdocument()
                daNote.Form="Group"
                daNote.ListName=nabNote.ListName(0)
            End If
            daNote.NABNoteUpdated=nabNote.lastmodified
            daNote.ListDescription=nabNote.ListDescription(0)
            If firstScan Then daNote.ParentGroup=nabNote.ListName(0)
            Set item=nabNote.Getfirstitem("Members")
            If item Is Nothing Then
                ' the Group in the NAB has novalues in the "Members" field. Just save the DA Group docum
                daNote.Members=""
                Call daNote.Save(t,t,t)
                Observation 1007,|The Deny Access Group named "|+daGroups(x)+|" has no Members lis
                LogStep norm,CStr(Now)+tb+|The Deny Access Group named "|+daGroups(x)+|" has no M
            ElseIf item.Text="" Then
                ' the Group in the NAB has novalues in the "Members" field. Just save the DA Group docum
                daNote.Members=""
                Call daNote.Save(t,t,t)
                Observation 1007,|The Deny Access Group named "|+daGroups(x)+|" has no Members lis
                LogStep norm,CStr(Now)+tb+|The Deny Access Group named "|+daGroups(x)+|" has no M
            Else
                ' cycle through all the "Members values and add them to the "member names" array
                z=0
                w=0
                ReDim memberNames(z)
                ReDim subGroups(0)
                For y=0 To UBound(item.Values)
                    ' If Not isGroup(CStr(item.Values(y)),allGroupNames,foundGroups) Then
                    If Not isGroup(LCase(CStr(item.Values(y))),foundGroups) Then
                        Set memberName=New NotesName(CStr(item.Values(y)))
                        If Not IsNumeric(ArrayGetIndex(memberNames,memberName.Abbreviated)) Then

```

```

        ReDim Preserve memberNames(z)
        memberNames(z)=memberName.Abbreviated
        z=z+1
    End If
Else
    ReDim Preserve subGroups(w)
    subGroups(w)=item.Values(y)
    w=w+1
End If
Next
Call daNote.Removeitem("Members")
Call daNote.Removeitem("SubGroups")
Call daNote.Replaceitemvalue("Members",FullTrim(memberNames))
Call daNote.Replaceitemvalue("SubGroups",FullTrim(subGroups))
If memberNames(0)="" Then daNote.MemberQuantity=0 Else daNote.MemberQuantity=UBound(memberNames)
If subGroups(0)="" Then daNote.SubgroupQuantity=0 Else daNote.SubgroupQuantity=UBound(subGroups)
Call daNote.Save(t,t,t)
End If
End If
' if any sub groups were located, add them to the list of all sub groups
If subGroups(0)<>"" Then
    LogStep norm,CStr(Now)+tb+|Processing Recursive Groups|
    z=UBound(allSubGroups)
    If allSubGroups(z)<>"" Then z=z+1
    For y=0 To UBound(subGroups)
        ' get the lowercase values of the "allSubGroups" array
        checkVal=LCase(Join(allSubGroups,"~"))
        If InStr(1,checkVal,subGroups(y),5)=0 Then
            ' group is not in the array yet, add it
            ReDim Preserve allSubGroups(z)
            allSubGroups(z)=subGroups(y)
            z=z+1
        End If
    Next
End If
Next
End If
Next
' if there were any sub groups located, start this process again
If allSubGroups(0)<>"" Then
    y=0
    For x=0 To UBound(allSubGroups)
        ReDim Preserve daGroups(y)
        daGroups(y)=allSubGroups(x)
        y=y+1
    Next
    ReDim allSubGroups(0)
    firstScan=f
    GoTo startScan
End If
' all DA Groups have been captured, now get the list of DA Groups used per server and store it back onto the table
daView.Autoupdate=t
Call daView.Refresh()
daView.Autoupdate=f
Dim newServerArray As Variant
ReDim newServerArray(0)
newServerArray(0)=""

```

```

serverArray=GetAdminValue("ServerList")
%REM
WHAT WE NEED TO DO NOW
1. Get all "Top Level" groups based on what is stored on the Server documents (In the "ServerList" Field On the
2. Get all Sub Group docs that are on the "Top Level" DA Group docs and stamp the "ParentGroup" value as the
   - also move the list of Servers from the Top Level Group onto the SubGroup
3. Repeat #2 for any SubGroups found on the SubGroups
%endrem
' first find all "Top Level" DA Groups that are listed on the Server docs and stamp them with the servers that
For x=0 To UBound(serverArray)
    If InStr(1,serverArray(x),"Available",5)<>0 Then ' server is available, get the DA Group doc and do the
        splitVar=Split(serverArray(x),"|")
        Set serverName=New NotesName(splitVar(0))
        splitVar=StrToken(serverArray(x),"|",3)
        If splitVar<>"" Then ' DA Groups are present, go get the DA Group names
            ReDim DAGroups(0)
            z=0
            If InStr(splitVar,"~") Then
                splitVar=Split(splitVar,"~")
                For y=0 To UBound(splitVar)
                    If Not IsNumeric(ArrayGetIndex(daGroups,splitVar(y))) Then
                        ReDim Preserve daGroups(z)
                        daGroups(z)=splitVar(y)
                        z=z+1
                    End If
                Next
            Else
                daGroups(z)=splitVar
                z=z+1
            End If
        ' now go get the DA Groups and stamp the server name onto them and also indicate that it is a
        For z=0 To UBound(daGroups)
            Set daNote=daView.Getdocumentbykey(daGroups(z))
            If Not daNote Is Nothing Then
                daNote.TopLevel="true"
                If daNote.hasitem("Servers") Then
                    newServerArray=daNote.Getfirstitem("Servers").Values
                    If Not IsNumeric(ArrayGetIndex(newServerArray,serverName.Abbreviated)) Then
                        w=UBound(newServerArray)
                        If newServerArray(w)<>"" Then w=w+1
                        ReDim Preserve newServerArray(w)
                        newServerArray(w)=serverName.Abbreviated
                        Call daNote.Removeitem("Servers")
                        Call daNote.Replaceitemvalue("Servers", FullTrim(newServerArray))
                    End If
                Else
                    daNote.Servers=serverName.Abbreviated
                End If
                Call daNote.Save(t,t,t)
            End If
        Next
    End If
Next
' NOTE - need to do a check to see if any of the servers DO NOT use Deny Access Groups, as this negates
' not sure 100% if this is needed, since this would also be picked up when processing the user docs (since

```



```

' Now, using the "Top Level" Groups, find any SubGroups that are listed on them, then find those SubGroup
Dim topDAView As NotesView
Dim subDAView As NotesView
Dim procView As NotesView
Dim unprocView As NotesView
Dim subNote As NotesDocument
Dim parentGroups() As String
Dim restart As Boolean
Set topDAView=db.Getview("parents")
Call topDAView.Refresh()
Set subDAView=db.Getview("subgroups")
Call subDAView.Refresh()
Set daNote=topDAView.Getfirstdocument()
While Not daNote Is Nothing
    If daNote.Hasitem("SubGroups") Then
        If daNote.SubGroups(0)<>"" Then
            For x=0 To UBound(daNote.SubGroups)
                Set subNote=subDAView.Getdocumentbykey(daNote.SubGroups(x))
                If Not subNote Is Nothing Then
                    ' sub group was found - stamp it with the Top Level Group name (in the ParentGroup fi
                    If subNote.Hasitem("ParentGroup") Then
                        If subNote.ParentGroup(0)<>"" Then
                            z=0
                            For y=0 To UBound(subNote.ParentGroup)
                                If LCase(subNote.ParentGroup(y))<>LCase(daNote.ListName(0)) Then
                                    ReDim Preserve parentGroups(z)
                                    parentGroups(z)=subNote.ParentGroup(y)
                                    z=z+1
                                End If
                            Next
                            ReDim Preserve parentGroups(z)
                            parentGroups(z)=daNote.ListName(0)
                        Else
                            subNote.ParentGroup=daNote.ListName(0)
                        End If
                    Else
                        subNote.ParentGroup=daNote.ListName(0)
                    End If
                    ' now add the list of servers from the Top Level Group onto the SubGroup (merge serv
                    Set item=daNote.Getfirstitem("Servers")
                    If subNote.Hasitem("Servers") Then
                        If subNote.Servers(0)<>"" Then
                            serverArray=ArrayAppend(item.Values,subnote.Getfirstitem("Servers").Values)
                            RemoveDuplicates serverArray
                            Call subNote.Removeitem("Servers")
                            Call subNote.Replaceitemvalue("Servers",serverArray)
                        Else
                            Call item.Copyitemtodocument(subNote,"Servers")
                        End If
                    Else
                        Call item.Copyitemtodocument(subNote,"Servers")
                    End If
                    Call subNote.Save(t,t,t)
                End If
            Next
        End If
    End If

```

```

        End If
        Set daNote=topDAView.Getnextdocument(daNote)
    Wend
loopStart:
    Set unprocView=db.Getview("sgunprocessed")
    Call unprocView.Refresh()
    unprocView.Autoupdate=f
    Call subDAView.Refresh()
    Set daNote=unprocView.Getfirstdocument()
    While Not daNote Is Nothing
        If daNote.Hasitem("SubGroups") Then
            If daNote.SubGroups(0)<>"" Then
                For x=0 To UBound(daNote.SubGroups)
                    Set subNote=subDAView.Getdocumentbykey(daNote.SubGroups(x))
                    If Not subNote Is Nothing Then
                        ' sub group was found - stamp it with the Top Level Group name (in the ParentGroup field)
                        If subNote.Hasitem("ParentGroup") Then
                            If subNote.ParentGroup(0)<>"" Then
                                z=0
                                For y=0 To UBound(subNote.ParentGroup)
                                    If LCase(subNote.ParentGroup(y))<>LCase(daNote.ListName(0)) Then
                                        ReDim Preserve parentGroups(z)
                                        parentGroups(z)=subNote.ParentGroup(y)
                                        z=z+1
                                    End If
                                Next
                                ReDim Preserve parentGroups(z)
                                parentGroups(z)=daNote.ListName(0)
                            Else
                                subNote.ParentGroup=daNote.ListName(0)
                            End If
                        Else
                            subNote.ParentGroup=daNote.ListName(0)
                        End If
                        ' now add the list of servers from the Top Level Group onto the SubGroup (merge servers)
                        Set item=daNote.Getfirstitem("Servers")
                        If subNote.Hasitem("Servers") Then
                            If subNote.Servers(0)<>"" Then
                                serverArray=ArrayAppend(item.Values,subnote.Getfirstitem("Servers").Values)
                                RemoveDuplicates serverArray
                                Call subNote.Removeitem("Servers")
                                Call subNote.Replaceitemvalue("Servers",serverArray)
                            Else
                                Call item.Copyitemtodocument(subNote,"Servers")
                            End If
                        Else
                            Call item.Copyitemtodocument(subNote,"Servers")
                        End If
                        Call subNote.Save(t,t,t)
                    End If
                Next
                daNote.Processed="true"
                Call daNote.Save(t,t,t)
            Else
                daNote.Processed="true"
                Call daNote.Save(t,t,t)
            End If
        End While
    End While

```

```

        End If
    Else
        daNote.Processed="true"
        Call daNote.Save(t,t,t)
    End If
    Set daNote=unprocView.Getnextdocument(daNote)
Wend
unprocView.Autoupdate=t
Call unprocView.Refresh()
Set daNote=unprocView.Getfirstdocument()
If Not daNote Is Nothing Then GoTo loopStart
' now count the # of DA Groups that are in this db and set that value to the admin doc
daView.Autoupdate=t
Call daView.Refresh()
daView.Autoupdate=f
Dim viewNav As NotesViewNavigator
Set viewNav=daView.Createviewnav()
SetAdminValue "GroupDocCount",viewNav.Count
If Not ProcessDenyAccessNames Then SetAdminValue "DAUserProcessingError","There was a problem
' indicate on the config doc if there is a server that does not have a Deny Access Group listed
If missingDA Then SetAdminValue"DAMissing","true" Else SetAdminValue "DAMissing",""
LogStep norm,CStr(Now)+tb+|"GetDenyAccessGroups" - finished|
GetDenyAccessGroups=t
errHandle:
If Err<>0 Then
    errString=Error
    errNum=Err
    errLine=Erl
    ReportError "There was an Error during the (GetDenyAccessGroups) function. The details of the Error
End If
Exit Function
End Function
Sub GetListOfOrgs(currentNAB As NotesDatabase)
On Error GoTo errHandle:
    Dim orgArray As Variant
    Dim x As Integer, y As Integer
    Dim formula As String
    Dim colVar As Variant
    orgArray=GetAdminValue("Orgs")
    formula="@DbColumn(;;"+currentNAB.Server+"::"+currentNAB.FilePath+";"PeopleCat";1)|
    colVar=Evaluate(formula)
    x=UBound(orgArray)
    If orgArray(x)<>"" Then x=x+1
    If colVar(0)<>"" Then
        For y=0 To UBound(colVar)
            If colVar(y)<>mainO Then
                If Not IsNumeric(ArrayGetIndex(orgArray,colVar(y))) Then
                    ReDim Preserve orgArray(x)
                    orgArray(x)=colVar(y)
                End If
            End If
        Next
    End If
    SetAdminValue "Orgs",orgArray
errHandle:
    Err=0

```

```

Exit Sub
End Sub
Function GetAdminServerInfo () As Boolean
    If Not debug Then On Error GoTo errHandle
    LogStep norm,CStr(Now)+tb+|"GetAdminServerInfo" - begin|
    Dim profile As NotesDocument
    Dim books As Variant
    Dim adminServer As String
    If session.Is Nothing Then QO
    ' find the server Domino Directory and discover the Admin Server value form the "DirectoryProfile" profile d
    books = session.AddressBooks
    ForAll b In books
        Call b.Open( "", "" )
        Set nab=b
        If nab.Server<>"" And LCase(nab.Filename)="names.nsf" Then
            Set profile=nab.Getprofiledocument("DirectoryProfile")
            If profile.Hasitem("AdminServer") Then
                If profile.AdminServer(0)<>"" Then
                    ' admin server has been located. set the value and exit the forall
                    adminServer=profile.AdminServer(0)
                    LogStep norm,CStr(Now)+tb+|In "GetAdminServerInfo" - svrName = |+adminServer
                    Exit ForAll
                End If
            End If
        End If
    End ForAll
    ' if the adminServer variable is not set, ask the user to select their Admin server's Domino Directory
    If adminServer="" Then SelectNAB adminServer
    ' if the adminServer value is still not set, exit out of this and set the value "Critical" onto the config document
    If adminServer="" Then SetAdminValue "Critical","true":Exit Function
    ' set the global "svrName" variable and then get the Admin server's directory
    Set svrName=New NotesName(adminServer)
    Set nab=session.Getdatabase(svrName.Canonical,"names.nsf")
    If Not nab.Isopen Then Call nab.Open("", "")
    If Not nab.isopen Then
        Set nab=session.Getdatabase(svrName.Canonical,nab.Filename)
        If Not nab.Isopen Then Call nab.Open("", "")
    End If
    If nab.Isopen Then
        SetAdminValue "AdminServer",svrName.Abbreviated
        GetListOfOrgs nab
    Else
        GoTo errNoNAB
    End If
    ' get the list of all Domino servers
    LogStep norm,CStr(Now)+tb+|"GetAdminServerInfo" - GetServerList - begin|
    SetAdminValue "ServerList",GetServerList
    LogStep norm,CStr(Now)+tb+|"GetAdminServerInfo" - GetServerList - finished|
    ' check the users ability to view the "Deny Access Groups" view. Without this access, the remainder of this
    LogStep norm,CStr(Now)+tb+|"GetAdminServerInfo" - CheckUserAccess - begin|
    If Not CheckUsersAccess Then SetAdminValue "Critical","true":Exit Function
    LogStep norm,CStr(Now)+tb+|"GetAdminServerInfo" - CheckUserAccess - finished|
    LogStep norm,CStr(Now)+tb+|"GetAdminServerInfo" - finished|
    GetAdminServerInfo=t
    Exit Function
errNoNAB:

```

```

MsgBox "Your Primary Domino Directory could not be located."+cr+cr+"Make sure that you can connect to
SetAdminValue "Critical","true"
Exit Function
errHandle:
If Err<>0 Then
    errString=Error
    errNum=Err
    errLine=Erl
    ReportError "There was an Error in the (GetAdminServerInfo) function. The details of the Error are below"
End If
Exit Function
End Function
Function IsGroupOLD(memberValue As String,allGroupNames() As String,foundGroups() As String) As Boolean
    Dim c As Integer
    If Not IsNumeric(ArrayGetIndex(allGroupNames,LCase(memberValue))) Then Exit Function ' the member value is not in the list
    c=UBound(foundGroups)
    If foundGroups(c)<>"" Then c=c+1
    ReDim Preserve foundGroups(c)
    foundGroups(c)=memberValue
    IsGroupOLD=t
End Function
Sub CheckLicenseTrackingSettings()
On Error GoTo errHandle
    Dim nabView As NotesView
    Dim nabNote As NotesDocument
    Dim adminVar As Variant
    Dim svrList As Variant
    Dim svrListNormalized As Variant
    Dim svrEnabledList() As String
    Dim svrDisabledList() As String
    Dim splitVar As Variant
    Dim svrName As NotesName
    Dim x As Integer, y As Integer, z As Integer, pos As Integer
    Dim globalEnabled As Boolean
    ReDim svrList(0)
    ReDim svrListNormalized(0)
    ReDim svrEnabledList(0)
    ReDim svrDisabledList(0)
    adminVar=GetAdminValue("ServerList")
    ' get a list of all Available servers
    For x=0 To UBound(adminVar)
        splitVar=Split(adminVar(x),",")
        If splitVar(1)="Available" Then
            ReDim Preserve svrListNormalized(y)
            ReDim Preserve svrList(y)
            svrListNormalized(y)=LCase(splitVar(0))
            svrList(y)=splitVar(0)
            y=y+1
        End If
    Next
    ' get the "Configurations" view in the domino Directory
    Set nabView=nab.Getview("Configurations")
    Call nabView.Refresh()
    nabView.Autoupdate=f
    ' get the first document in the view and check if it is the "General Configuration Settings" doc, if it is then check the license
    Set nabNote=nabView.Getfirstdocument()

```

```

If nabNote Is Nothing Then GoTo notEnabled
If nabNote.ServerName(0)="" Then
    If nabNote.HasItem("LicTrackON") Then If nabNote.LicTrackON(0)="1" Then globalEnabled=t
End If
If globalEnabled Then
    Set nabNote=nabView.Getnthdocument(2)
Else
    Set nabNote=nabView.Getfirstdocument()
End If
x=0
y=0
While Not nabNote Is Nothing
    Set svrName=New NotesName(nabNote.ServerName(0))
    If IsNumeric(ArrayGetIndex(svrListNormalized,LCase(svrName.Abbreviated))) Then ' server i
        Call nabNote.Computewithform(f,f)
        If nabNote.LicTrackON(0)="1" Then
            ReDim Preserve svrEnabledList(x)
            svrEnabledList(x)=svrName.Abbreviated+"|"+svrName.Abbreviated
            x=x+1
        Else
            ReDim Preserve svrDisabledList(y)
            svrDisabledList(y)=svrName.Abbreviated
            y=y+1
        End If
        ' remove the server name from the full list of servers
        pos=ArrayGetIndex(svrListNormalized,LCase(svrName.Abbreviated))
        svrList(pos)=""
        svrListNormalized(pos)=""
    End If
    Set nabNote=nabView.Getnextdocument(nabNote)
Wend
notEnabled:
svrList=FullTrim(svrList) ' remove any blank entries in the list of servers
If svrList(0)<>"" Then
    If globalEnabled Then ' if the global config has the value enabled, add the remaining servers
        x=UBound(svrEnabledList)
        If svrEnabledList(x)<>"" Then x=x+1
        For y=0 To UBound(svrList)
            ReDim Preserve svrEnabledList(x)
            svrEnabledList(x)=svrList(y)+"|"+svrList(y)
            x=x+1
        Next
    Else ' if the global config has the value disabled, add the remain
        x=UBound(svrDisabledList)
        If svrDisabledList(x)<>"" Then x=x+1
        For y=0 To UBound(svrList)
            ReDim Preserve svrDisabledList(x)
            svrDisabledList(x)=svrList(y)+"|"+svrList(y)
            x=x+1
        Next
    End If
End If
' for the servers that are configured, set their names into the "LicenseTrackingServers" config field
SetAdminValue "LicenseTrackingServers",svrEnabledList
' for the servers that are NOT configured, set their names into the "LicenseTrackingServersMissing" config
SetAdminValue "LicenseTrackingServersMissing",svrDisabledList

```

```

' for all servers in the "disabled" list, add an Observation
If svrDisabledList(0)<>"" Then
    For x=0 To UBound(svrDisabledList)
        Observation 1004,"The License Tracking task is not running on the HCL Domino server "+svrDisab
        LogStep norm,CStr(Now)+tb+|The License Tracking task is not running on the HCL Domino server
    Next
End If
errHandle:
If Err<>0 Then
    errString=Error
    errNum=Err
    errLine=Erl
    ReportError "There was an Error in the (CheckLicenseTrackingSettings) function. The details of the Err
End If
Exit Sub
End Sub
Function CheckDesign() As Boolean
If Not debug Then On Error GoTo errHandle
Dim view As NotesView
Dim viewNote As NotesDocument
Dim valCheck As NotesName
If db Is Nothing Then QO
Set view=db.Getview("config")
If view Is Nothing Then
    errNum=9001
    errLine=12
    errString=|Design element cannot be located - View: (FormDesign)|
    GoTo errHandle
End If
Set viewNote=db.Getdocumentbyunid(view.Universalid)
Set valCheck=New NotesName(viewNote.Getfirstitem("$UpdatedBy").Values(UBound(viewNote.Getfirstite
If valCheck.Organization<>currentUser.Organization Then Exit Function
CheckDesign=t
errHandle:
If Err<>0 Then
    errString=Error
    errNum=Err
    errLine=Erl
    ReportError "There was an Error checking the signer of this HCL Notes application design (CheckDesi
End If
Exit Function
End Function
Function CheckUsersAccess() As Boolean
' in this function, if anything fails exit and do not allow user to continue with any additional functionality
If Not debug Then On Error GoTo errHandle
LogStep norm,CStr(Now)+tb+|"CheckUserAccess" - begin|
Dim aclLevel As Integer
Dim checkView As NotesView
' check the user's ACL access. If it is less than Designer, alert the user and exit this
aclLevel=nab.Currentaccesslevel
If aclLevel<5 Then GoTo exitWithMsg
SetProfileValue userProfileName,session.Username,"AccessLevel",aclLevel
' check to see if the user can access the "Deny Access Groups" in the nab. If they can see it hten they have
Set checkView=nab.Getview("DenyLists")
If checkView Is Nothing Then GoTo exitNoRole
SetProfileValue userProfileName,session.Username,"HasRole","true"

```

```

        SetProfileValue userProfileName,session.Username,"HasAccess","true"
        LogStep norm,CStr(Now)+tb+"CheckUserAccess" - finished|
        CheckUsersAccess=t
        Exit Function
    exitWithMsg:
        MsgBox "You do not have the proper access level to the HCL Domino Directory on the server: "+GetAdminValue
        Exit Function
    exitNoRole:
        MsgBox |You do not have the proper access to the HCL Domino Directory on the server: |+GetAdminValue
        Exit Function
    errHandle:
        ' some errors cannot be trapped. For this reason, we are not alerting the user to any of the errors, we're just
        Exit Function
End Function
Function ProcessEntitlements(processDB As NotesDatabase, isUserReport As Boolean) As Boolean
    If Not debug Then On Error GoTo errHandle
    LogStep norm,CStr(Now)+tb+"ProcessEntitlements" - begin|
    Dim enView As NotesView
    Dim enNote As NotesDocument
    Dim enNav As NotesViewNavigator
    Dim userNote As NotesDocument
    Dim userView As NotesView
    Dim item As NotesItem
    Dim perName As NotesName
    Dim enServer As NotesName
    Dim key As String
    Dim nameCol As Integer
    Dim serverCol As Integer
    Dim accessCol As Integer
    Dim dbCol As Integer
    Dim aclCol As Integer
    Dim newVal As String
    Dim valArray() As String
    Dim pos As Integer
    Dim x As Integer,y As Integer
    ReDim valArray(0)
    nameCol=0
    accessCol=1
    aclCol=3
    If isUserReport Then
        serverCol=1
        accessCol=2
        dbCol=3
        aclCol=4
    End If
    Set enView=processDB.Getview("Users")
    Call enView.Refresh()
    enView.Autoupdate=f
    Dim svrName As New NotesName(processDB.Server)
    Set userView=db.Getview("users")
    Call userView.Refresh()
    userView.Autoupdate=f
    Set enNav=enView.Createviewnav()
    If enNav.Count=0 Then GoTo exitThis
    Set userNote=userView.Getfirstdocument()
    While Not userNote Is Nothing

```



```

Set perName=New NotesName(userNote.FullName(0))
key=perName.Abbreviated
Set enNote=enView.Getdocumentbykey(key)
If Not enNote Is Nothing Then
    If isUserReport Then
        newVal=enNote.Columnvalues(serverCol)
    Else
        Set enServer=New NotesName(enNote.HighestAccessServerName(0))
        newVal=enServer.Abbreviated
    End If
    key=newVal
    If isUserReport Then
        newVal=newVal+"~"+enNote.Columnvalues(accessCol)+"~"+enNote.Columnvalues(dbCol)+"~"
    Else
        newVal=newVal+"~"+enNote.Columnvalues(accessCol)+"~"+enNote.HighestAccessDatabaseName(0)+"~"
    End If
    If userNote.Hasitem("EntitlementList") Then
        pos=-1
        ReDim valArray(UBound(userNote.EntitlementList))
        For x=0 To UBound(userNote.EntitlementList)
            valArray(x)=userNote.EntitlementList(x)
            If InStr(1,userNote.EntitlementList(x),key,5)<>0 Then pos=x
        Next
        If pos>-1 Then
            valArray(pos)=newVal
        Else
            x=UBound(valArray)+1
            ReDim Preserve valArray(x)
            valArray(x)=newVal
        End If
    Else
        ReDim valArray(0)
        valArray(0)=newVal
    End If
    Call userNote.RemoveItem("EntitlementList")
    Call userNote.ReplaceItemValue("EntitlementList",valArray)
    Call userNote.Save(t,t,t)
End If
Set userNote=userView.getnextdocument(userNote)
Wend
exitThis:
LogStep norm,CStr(Now)+tb+"ProcessEntitlements" - finished|
ProcessEntitlements=t
Exit Function
errHandle:
If Err<>0 Then
    errString=Error
    errNum=Err
    errLine=Erl
    ReportError "There was an Error in the (ProcessEntitlements) function. The details of the Error are below"
End If
Exit Function
End Function
Function GetServerList() As Variant
    On Error Resume Next
    Dim svrView As NotesView

```

```

Dim nav As NotesViewNavigator
Dim entry As NotesViewEntry
Dim entryNote As NotesDocument
Dim testDB As NotesDatabase
Dim svrName As NotesName
Dim count As Integer
Dim tmpVal As Variant
Dim availCount As Integer
Dim httpSecure As Boolean
Dim ldapSecure As Boolean
Dim valSet As String
Dim colNum As Integer
Dim online As Boolean
' assume both of these will be true unless found otherwise when scanning the server document
httpSecure=t
ldapSecure=t
Set svrView=nab.GetView("($Servers)")
If svrView Is Nothing Then
    errString=|A View is missing form your Domino Directory - "($Servers)"|
    errNum=9999
    errLine=35
    errCritical=t
    ReportError "There was an Error in the (GetServerList) function. The details of the Error are below:"
    Exit Function
End If
svrView.AutoUpdate = False
Set nav = svrView.CreateViewNav
Set entry = nav.GetFirstDocument
While Not entry Is Nothing
    If IsArray (entry.ColumnValues) Then
        If entry.ColumnValues (0)<>"" And entry.Columnvalues(0)<>"" Then
            ReDim Preserve serverList (count)
            Set entryNote=entry.Document
            Set svrName=New NotesName(entryNote.ServerName(0))
            serverList(count)=svrName.Abbreviated
            ' check to see if the server is available by attempting access to it's Domino Log file
            Set testDB=session.Getdatabase(svrName.Canonical, "log.nsf")
            If Not testDB.Isopen Then Call testDB.Open("", "")
            If testDB.Isopen Then
                serverList(count)=serverList(count)+"|Available"
                availCount=availCount+1
                online=t
            Else
                serverList(count)=serverList(count)+"|Offline"
                Observation 1001,|The HCL Domino Server "|+svrName.Abbreviated+|" appears to be Offl
                LogStep warn,CStr(Now)+tb+|The HCL Domino Server "|+svrName.Abbreviated+|" appear
                online=f
            End If
            ' Get the list of Deny Access values from the server document
            tmpVal=Evaluate(|@Implode(DenyAccess;"~")|,entryNote)
            serverList(count)=serverList(count)+"|"+tmpVal(0)
            serverList(count)=serverList(count)+"|"
            If entryNote.HTTP_NormalMode(0)="0" Then
                serverList(count)=serverList(count)+"true"
                ' if the port is not
            Elseif entryNote.HTTP_EnforceAccess(0)="1" Then
                serverList(count)=serverList(count)+"true"
                ' if the port is enforcing Deny Acco
        End If
    End While

```

```

Else
    If online Then
        If testDB.isopen Then httpSecure=f
        Observation 1002,|The HCL Domino Server "|+svrName.Abbreviated+|" is not properly
        LogStep sec,CStr(Now)+tb+|The HCL Domino Server "|+svrName.Abbreviated+|" is n
    End If
End If
serverList(count)=serverList(count)+"|"
If entryNote.LDAP_PortStatus(0)="0" Then ' if the port is not
    serverList(count)=serverList(count)+"true"
ElseIf entryNote.LDAP_EnforceAccess(0)="1" Then ' if the port is enforcing Deny Acco
    serverList(count)=serverList(count)+"true"
Else
    If online Then
        If testDB.isopen Then ldapSecure=f
        Observation 1003,|The HCL Domino Server "|+svrName.Abbreviated+|" is not properly
        LogStep sec,CStr(Now)+tb+|The HCL Domino Server "|+svrName.Abbreviated+|" is n
    End If
End If
count = count + 1
End If
End If
Err=0
Set entry=nav.GetNextDocument (entry)
Wend
valSet="false"
If httpSecure Then valSet="true"
SetAdminValue "HTTPSecure",valSet
valSet="false"
If ldapSecure Then valSet="true"
SetAdminValue "LDAPSecure",valSet
SortArray serverList
GetServerList=serverList
Err=0
End Function
Function GetDADirectories() As Boolean
    If Not debug Then On Error GoTo errHandle
    LogStep norm,CStr(Now)+tb+|"GetDADirectories" - begin|
    Dim daDC As NotesDocumentCollection
    Dim daNote As NotesDocument
    Dim rti As NotesRichTextItem
    Dim rtnav As NotesRichTextNavigator
    Dim rtlink As NotesRichTextDocLink
    Dim daArray() As String
    Dim fullArray() As String
    Dim x As Integer, y As Integer
    Dim splitVar As Variant
    Dim testDB As NotesDatabase
    Dim dblInfo As String
    Dim adminServer As String
    ReDim daArray(0)
    ReDim fullArray(0)
    adminServer=GetAdminValue(" AdminServer")(0)
    Set daDC=dadb.Search(|Form="DirectoryAssistance"|,Nothing , 0)
    If daDC.count=0 Then
        SetAdminValue "DADirectoryList","< No Additional Domino Directories are configured in the Directory

```

```

        GoTo exitThis
    End If
    Set daNote=daDC.Getfirstdocument()
    While Not daNote Is Nothing
        If daNote.DomainType(0)="LDAP" Then
            Observation 1009,|The Directory Assistance configuration "|+daNote.DomainName(0)+|" (|+daNote
            SetAdminValue "LDAPDA","true"
        Else
            Set rti=daNote.Getfirstitem("DatabaseLinks")
            If Not rti Is Nothing Then
                Set rtnav=rti.Createnavigator()
                If rtnav.FindFirstElement(RTELEM_TYPE_DOCLINK) Then ' links have been found. process th
                    Do
                        Set rtlink = rtnav.GetElement
                        If rtlink.DbReplicaID<>"" Then
                            ReDim Preserve daArray(x)
                            daArray(x)=rtlink.Serverhint+"~"+rtlink.DbReplicaID
                            x=x+1
                        End If
                    Loop While rtnav.FindNextElement
                End If
            End If
            For y=0 To 4 ' loop through the pre-defined set of fields for manually listing additional directories
                If daNote.Getfirstitem("Enabled_"+CStr(y)).Text="1" And daNote.Getfirstitem("ServerName_ "+
                    ReDim Preserve daArray(x)
                    If daNote.Getfirstitem("ServerName_"+CStr(y)).Text="" Then
                        daArray(x)=adminServer+"~"+LCase(daNote.Getfirstitem("AddressBook_"+CStr(y)).T
                    Else
                        daArray(x)=daNote.Getfirstitem("ServerName_"+CStr(y)).Text+"~"+LCase(daNote.Ge
                    End If
                    x=x+1
                End If
            Next
        End If
        Set daNote=daDC.Getnextdocument(daNote)
    Wend
    If daArray(0)<>"" Then ' look for the extra directories. If they are available then list them
        y=0
        For x=0 To UBound(daArray)
            splitVar=Split(daArray(x),"~")
            If Right(splitVar(1),4)=".nsf" Then
                Set testDB=session.Getdatabase(splitVar(0), splitVar(1))
            Else
                Set testDB=New NotesDatabase("", "")
                Call testDB.OpenbyreplicaID(splitVar(0), splitVar(1))
            End If
            If Not testDB.isopen Then Call testDB.Open("", "")
            If testDB.Isopen Then
                Set svrName=New NotesName(testDB.Server)
                dblInfo=svrName.Abbreviated+"|"+testDB.Filepath
                LogStep norm,CStr(Now)+tb+|Checking design of Domino Directory - |+testDB.Title+" ("+dblInf
                If CheckNABDesign(testDB) Then
                    If Not IsNumeric(ArrayGetIndex(fullArray,testDB.Title+" ("+dblInfo+"")+ "~"+dblInfo)) Then
                        ReDim Preserve fullArray(y)
                        fullArray(y)=testDB.Title+" ("+dblInfo+"")+ "~"+dblInfo
                        GetListOfOrgs testDB
                    End If
                End If
            End If
        Next x
    Next y
    Exit This
End Sub

```

```

        y=y+1
    End If
    Else
        Observation 1008,|The HCL Domino Directory configured in Directory Assistance cannot b
        LogStep warn,CStr(Now)+tb+|The HCL Domino Directory configured in Directory Assistan
    End If
    Else
        LogStep norm,CStr(Now)+tb+|Invalid Domino Directory design, skipping - |+testDB.Title+" ("+c
    End If
    Next
    SetAdminValue "DAList",fullArray
End If
exitThis:
    LogStep norm,CStr(Now)+tb+|"GetDADirectories" - finished|
    GetDADirectories=t
errHandle:
    If Err<>0 Then
        errString=Error
        errNum=Err
        errLine=Err
        ReportError "There was an Error in the (GetDADirectories) function. The details of the Error are below:
    End If
    Exit Function
End Function
Sub SelectURT()
    Dim svr As NotesName
    Dim newList As Variant
    Dim newValue As String
    Dim dbFile As Variant
    Dim sourceInfo(1) As String
    Dim selectedDB As NotesDatabase
    Dim x As Integer
selectURT:
    If session Is Nothing Then QO
    If ws Is Nothing Then Set ws=New NotesUIWorkspace
    newList=ws.Prompt(13,"Select HCL Notes Application","Select the User Report Tool from the list below")
    If IsEmpty(newList) Then Exit Sub
    If UBound(newList)>0 Then ' let's see if they selected multiple dbs
        If UBound(newList)<>2 Then GoTo tryAgain
        If InStr(1,newList(1),"~")<>0 Then GoTo tryAgain
        If InStr(1,newList(1),|.nsf" "|,5)<>0 Then GoTo tryAgain
    End If
    If UBound(newList)=2 Then ' array - extract proper values
        Set svr=New NotesName(newList(0))
        If svr.Common="" Then sourceInfo(0)="Local Workstation" Else sourceInfo(0)=svr.Abbreviated
        sourceInfo(1)=newList(1)
    Else ' single value, split it up
        newValue=StrToken(newList(0),"|",2)
        Set svr=New NotesName(StrToken(newValue,"~",1))
        If svr.Common="" Then sourceInfo(0)="Local Workstation" Else sourceInfo(0)=svr.Abbreviated
        sourceInfo(1)=StrToken(newValue,"~",2)
    End If
    If InStr(1,sourceInfo(1),"\")<>0 Then
        dbFile=Split(Left(sourceInfo(1),Len(sourceInfo(1))-4),"\")
        dbFile=dbFile(UBound(dbFile))
    Else

```

```

        dbFile=Left(sourceInfo(1),Len(sourceInfo(1))-4)
    End If
    If sourceInfo(0)="Local Workstation" Then Set selectedDB=session.GetDatabase("",sourceInfo(1)) Else Set
    If Not selectedDB.IsOpen Then Call selectedDB.Open("", "")
    If Not selectedDB.IsOpen Then MsgBox "The User Report Tool you selected is not accessible. Please try a
    Dim checkForm As NotesForm
    Dim checkView As NotesView
    Set checkForm=selectedDB.Getform("Person")
    If checkForm Is Nothing Then GoTo notValid
    Set checkForm=selectedDB.Getform("Server")
    If checkForm Is Nothing Then GoTo notValid
    Set checkView=selectedDB.Getview("Server Summary")
    If checkView Is Nothing Then GoTo notValid
    Set checkView=selectedDB.Getview("Users")
    If checkView Is Nothing Then GoTo notValid
    Set checkView=selectedDB.Getview("AccessByAccess")
    If checkView Is Nothing Then GoTo notValid
    Dim value As String
    value=selectedDB.Title+" ("&sourceInfo(0)&"!!"&sourceInfo(1)&")|"&sourceInfo(0)&"~"&sourceInfo(1)
    SetAdminValue "UserReportLocation",value
    Exit Sub
notValid:
    If MsgBox("The HCL Notes application that you chose is not a valid User Report Tool,"&cr&cr&"Do you wan
    Exit Sub
tryagain:
    MsgBox "Only one application can be selected. Please try again."
    Exit Sub
End Sub
Function GetLicenseTrackingInfo(licDB As NotesDatabase, noteCount As Long) As Boolean
    If Not debug Then On Error GoTo errHandle
    LogStep norm,CStr(Now)+tb+|"GetLicenseTrackingInfo" - begin|
    Dim licNote As NotesDocument
    Dim licView As NotesView
    Dim userNote As NotesDocument
    Dim userView As NotesView
    Dim userNav As NotesViewNavigator
    Dim userEntry As NotesViewEntry
    Dim userName As NotesName
    Dim item As NotesItem
    Dim copyItem As NotesItem
    Dim update As Boolean
    Dim tmpVal As Variant
    Dim unlicensed As Boolean
    Set licView=licDB.Getview("AllUsers")
    Call licView.Refresh()
    licView.Autoupdate=f
    Set userView=db.Getview("possibleusers")
    Call userView.Refresh()
    userView.Autoupdate=f
    Set userNav=userView.Createviewnav()
    Set userNote=userView.Getfirstdocument()
    Set userEntry=userNav.Getfirst()
    While Not userEntry Is Nothing
    '        unlicensed=f
    '        If userNote.hasitem("UserType") Then If userNote.UserType(0)="Unknown / Unlicensed" Then unlicen
        Set userName=New NotesName(userEntry.Columnvalues(0))

```

```

Set licNote=licView.Getdocumentbykey(userName.Abbreviated,t)
If Not licNote Is Nothing Then
    Set userNote=userEntry.Document
    Call userNote.Removeitem("LicenseProtocol")
    Call userNote.Removeitem("LicenseTimestamp")
    Call userNote.Removeitem("LicenseTimestampHTTP")
    Call userNote.Removeitem("LicenseTimestampLDAP")
    Call userNote.Removeitem("LicenseTimestampNotes")
    Set item=licNote.Getfirstitem("Server")
    Set copyitem=item.Copyitemtodocument(userNote, "LicenseServer")
    Set item=licNote.Getfirstitem("Protocol")
    Set copyitem=item.Copyitemtodocument(userNote, "LicenseProtocol")
    Set item=licNote.Getfirstitem("Timestamp")
    Set copyitem=item.Copyitemtodocument(userNote, "LicenseTimestamp")
    If licNote.Hasitem("TimestampHTTP") Then
        Set item=licNote.Getfirstitem("TimestampHTTP")
        Set copyitem=item.Copyitemtodocument(userNote, "LicenseTimestampHTTP")
        userNote.UserType="CAL"
    End If
    If licNote.Hasitem("TimestampLDAP") Then
        Set item=licNote.Getfirstitem("TimestampLDAP")
        Set copyitem=item.Copyitemtodocument(userNote, "LicenseTimestampLDAP")
        userNote.UserType="LDAP"
    End If
    If licNote.Hasitem("TimestampNotes") Then
        Set item=licNote.Getfirstitem("TimestampNotes")
        Set copyitem=item.Copyitemtodocument(userNote, "LicenseTimestampNotes")
        userNote.UserType="Client"
    End If
    If userNote.HasItem("NABNOTECERTEXPIRY") Then userNote.UserType="Client"
    Call userNote.Save(t,t,t)
    noteCount=noteCount+1
End If
Set userEntry=userNav.Getnext(userEntry)
Wend
SetAdminValue "LicenseTrackingDocCount",noteCount
LogStep norm,CStr(Now)+tb+"GetLicenseTrackingInfo" - finished|
GetLicenseTrackingInfo=t
errHandle:
If Err<>0 Then
    errString=Error
    errNum=Err
    errLine=Erl
    ReportError "There was an Error during the (GetLicenseTrackingInfo) function. The details of the Error"
End If
Exit Function
End Function
Function CheckNABDesign(currentNAB As NotesDatabase) As Boolean
    Dim forms(3) As String
    Dim subforms(3) As String
    Dim views(3) As String
    Dim searchForm As NotesForm
    Dim searchView As NotesView
    Dim c As Integer
    forms(0)="Person"
    forms(1)="Group"

```

```

forms(2)="Server"
forms(3)="WebSite"
subforms(0)="$ServerTasks"
subforms(1)="$InternetSite"
subforms(2)="$IMAP"
subforms(3)="$PersonGeneralInfo"
views(0)="People"
views(1)="Groups"
views(2)="Servers"
views(3)="Policies"
For c=0 To UBound(forms)
    Set searchForm=currentNAB.Getform(forms(c))
    If searchForm Is Nothing Then Exit Function
Next
For c=0 To UBound(subforms)
    Set searchForm=currentNAB.Getform(subforms(c))
    If searchForm Is Nothing Then Exit Function
Next
For c=0 To UBound(views)
    Set searchView=currentNAB.Getview(views(c))
    If searchView Is Nothing Then Exit Function
Next
CheckNABDesign=t
End Function
Sub GetAllGroupNames(groupView As NotesView,daGroupView As NotesView)
    Dim nabNote As NotesDocument
    Dim x As Integer
    Dim arrayCount As Integer
    Dim loadArray As Variant
    Const top=25000
    Dim loadStep As Integer
    Dim arrayNum As Integer
    ReDim allGroupNames1(0)
    ReDim allGroupNames2(0)
    ReDim allGroupNames3(0)
    ReDim allGroupNames4(0)
    ReDim allGroupNames5(0)
    ReDim allGroupNames6(0)
    ReDim allGroupNames7(0)
    ReDim allGroupNames8(0)
    ReDim allGroupNames9(0)
    ReDim allGroupNames10(0)
    ReDim allGroupNames11(0)
    ReDim allGroupNames12(0)
    ReDim allGroupNames13(0)
    ReDim allGroupNames14(0)
    ReDim allGroupNames15(0)
    ReDim allGroupNames16(0)
    ReDim allGroupNames17(0)
    ReDim allGroupNames18(0)
    ReDim allGroupNames19(0)
    ReDim allGroupNames20(0)
    ReDim allGroupNames21(0)
    ReDim allGroupNames22(0)
    ReDim allGroupNames23(0)
    ReDim allGroupNames24(0)

```



```
ReDim allGroupNames25(0)
ReDim allGroupNames26(0)
ReDim allGroupNames27(0)
ReDim allGroupNames28(0)
ReDim allGroupNames29(0)
ReDim allGroupNames30(0)
ReDim allGroupNames31(0)
ReDim allGroupNames32(0)
ReDim allGroupNames33(0)
ReDim allGroupNames34(0)
ReDim allGroupNames35(0)
ReDim allGroupNames36(0)
ReDim allGroupNames37(0)
ReDim allGroupNames38(0)
ReDim allGroupNames39(0)
ReDim allGroupNames40(0)
allGroupNames1(0)=""
allGroupNames2(0)=""
allGroupNames3(0)=""
allGroupNames4(0)=""
allGroupNames5(0)=""
allGroupNames6(0)=""
allGroupNames7(0)=""
allGroupNames8(0)=""
allGroupNames9(0)=""
allGroupNames10(0)=""
allGroupNames11(0)=""
allGroupNames12(0)=""
allGroupNames13(0)=""
allGroupNames14(0)=""
allGroupNames15(0)=""
allGroupNames16(0)=""
allGroupNames17(0)=""
allGroupNames18(0)=""
allGroupNames19(0)=""
allGroupNames20(0)=""
allGroupNames21(0)=""
allGroupNames22(0)=""
allGroupNames23(0)=""
allGroupNames24(0)=""
allGroupNames25(0)=""
allGroupNames26(0)=""
allGroupNames27(0)=""
allGroupNames28(0)=""
allGroupNames29(0)=""
allGroupNames30(0)=""
allGroupNames31(0)=""
allGroupNames32(0)=""
allGroupNames33(0)=""
allGroupNames34(0)=""
allGroupNames35(0)=""
allGroupNames36(0)=""
allGroupNames37(0)=""
allGroupNames38(0)=""
allGroupNames39(0)=""
allGroupNames40(0)=""
```

```

ReDim loadArray(0)
' load all available Group Names into an array for use later in this function
Set nabNote=groupView.Getfirstdocument()
step1:
While Not nabNote Is Nothing
    ReDim Preserve loadArray(x)
    loadArray(x)=LCase(nabNote.ListName(0))
    x=x+1
    Set nabNote=groupView.Getnextdocument(nabNote)
    If x=top Then
        loadStep=1
        GoTo loadGroupArray
    End If
Wend
Set nabNote=daGroupView.Getfirstdocument()
step2:
While Not nabNote Is Nothing
    ReDim Preserve loadArray(x)
    loadArray(x)=LCase(nabNote.ListName(0))
    x=x+1
    Set nabNote=daGroupView.Getnextdocument(nabNote)
    If x=top Then
        loadStep=2
        GoTo loadGroupArray
    End If
Wend
loadStep=3
loadGroupArray:
x=0
arrayNum=arrayNum+1
Select Case arrayNum
    Case 1: allGroupNames1=loadArray
    Case 2: allGroupNames2=loadArray
    Case 3: allGroupNames3=loadArray
    Case 4: allGroupNames4=loadArray
    Case 5: allGroupNames5=loadArray
    Case 6: allGroupNames6=loadArray
    Case 7: allGroupNames7=loadArray
    Case 8: allGroupNames8=loadArray
    Case 9: allGroupNames9=loadArray
    Case 10: allGroupNames10=loadArray
    Case 11: allGroupNames11=loadArray
    Case 12: allGroupNames12=loadArray
    Case 13: allGroupNames13=loadArray
    Case 14: allGroupNames14=loadArray
    Case 15: allGroupNames15=loadArray
    Case 16: allGroupNames16=loadArray
    Case 17: allGroupNames17=loadArray
    Case 18: allGroupNames18=loadArray
    Case 19: allGroupNames19=loadArray
    Case 20: allGroupNames20=loadArray
    Case 21: allGroupNames21=loadArray
    Case 22: allGroupNames22=loadArray
    Case 23: allGroupNames23=loadArray
    Case 24: allGroupNames24=loadArray
    Case 25: allGroupNames25=loadArray

```

```

        Case 26: allGroupNames26=loadArray
        Case 27: allGroupNames27=loadArray
        Case 28: allGroupNames28=loadArray
        Case 29: allGroupNames29=loadArray
        Case 30: allGroupNames30=loadArray
        Case 31: allGroupNames31=loadArray
        Case 32: allGroupNames32=loadArray
        Case 33: allGroupNames33=loadArray
        Case 34: allGroupNames34=loadArray
        Case 35: allGroupNames35=loadArray
        Case 36: allGroupNames36=loadArray
        Case 37: allGroupNames37=loadArray
        Case 38: allGroupNames38=loadArray
        Case 39: allGroupNames39=loadArray
        Case 30: allGroupNames40=loadArray
    End Select
    If loadStep=1 Then GoTo step1
    If loadStep=2 Then GoTo step2
    Exit Sub
End Sub
Function IsGroup(memberValue As String,foundGroups() As String) As Boolean
    Dim c As Integer
    If IsNumeric(ArrayGetIndex(allGroupNames1,memberValue)) Then IsGroup=t
    If IsNumeric(ArrayGetIndex(allGroupNames2,memberValue)) Then IsGroup=t
    If IsNumeric(ArrayGetIndex(allGroupNames3,memberValue)) Then IsGroup=t
    If IsNumeric(ArrayGetIndex(allGroupNames4,memberValue)) Then IsGroup=t
    If IsNumeric(ArrayGetIndex(allGroupNames5,memberValue)) Then IsGroup=t
    If IsNumeric(ArrayGetIndex(allGroupNames6,memberValue)) Then IsGroup=t
    If IsNumeric(ArrayGetIndex(allGroupNames7,memberValue)) Then IsGroup=t
    If IsNumeric(ArrayGetIndex(allGroupNames8,memberValue)) Then IsGroup=t
    If IsNumeric(ArrayGetIndex(allGroupNames9,memberValue)) Then IsGroup=t
    If IsNumeric(ArrayGetIndex(allGroupNames10,memberValue)) Then IsGroup=t
    If IsNumeric(ArrayGetIndex(allGroupNames11,memberValue)) Then IsGroup=t
    If IsNumeric(ArrayGetIndex(allGroupNames12,memberValue)) Then IsGroup=t
    If IsNumeric(ArrayGetIndex(allGroupNames13,memberValue)) Then IsGroup=t
    If IsNumeric(ArrayGetIndex(allGroupNames14,memberValue)) Then IsGroup=t
    If IsNumeric(ArrayGetIndex(allGroupNames15,memberValue)) Then IsGroup=t
    If IsNumeric(ArrayGetIndex(allGroupNames16,memberValue)) Then IsGroup=t
    If IsNumeric(ArrayGetIndex(allGroupNames17,memberValue)) Then IsGroup=t
    If IsNumeric(ArrayGetIndex(allGroupNames18,memberValue)) Then IsGroup=t
    If IsNumeric(ArrayGetIndex(allGroupNames19,memberValue)) Then IsGroup=t
    If IsNumeric(ArrayGetIndex(allGroupNames20,memberValue)) Then IsGroup=t
    If IsNumeric(ArrayGetIndex(allGroupNames21,memberValue)) Then IsGroup=t
    If IsNumeric(ArrayGetIndex(allGroupNames22,memberValue)) Then IsGroup=t
    If IsNumeric(ArrayGetIndex(allGroupNames23,memberValue)) Then IsGroup=t
    If IsNumeric(ArrayGetIndex(allGroupNames24,memberValue)) Then IsGroup=t
    If IsNumeric(ArrayGetIndex(allGroupNames25,memberValue)) Then IsGroup=t
    If IsNumeric(ArrayGetIndex(allGroupNames26,memberValue)) Then IsGroup=t
    If IsNumeric(ArrayGetIndex(allGroupNames27,memberValue)) Then IsGroup=t
    If IsNumeric(ArrayGetIndex(allGroupNames28,memberValue)) Then IsGroup=t
    If IsNumeric(ArrayGetIndex(allGroupNames29,memberValue)) Then IsGroup=t
    If IsNumeric(ArrayGetIndex(allGroupNames30,memberValue)) Then IsGroup=t
    If IsNumeric(ArrayGetIndex(allGroupNames31,memberValue)) Then IsGroup=t
    If IsNumeric(ArrayGetIndex(allGroupNames32,memberValue)) Then IsGroup=t
    If IsNumeric(ArrayGetIndex(allGroupNames33,memberValue)) Then IsGroup=t
    If IsNumeric(ArrayGetIndex(allGroupNames34,memberValue)) Then IsGroup=t

```

```

If IsNumeric(ArrayGetIndex(allGroupNames35,memberValue)) Then IsGroup=t
If IsNumeric(ArrayGetIndex(allGroupNames36,memberValue)) Then IsGroup=t
If IsNumeric(ArrayGetIndex(allGroupNames37,memberValue)) Then IsGroup=t
If IsNumeric(ArrayGetIndex(allGroupNames38,memberValue)) Then IsGroup=t
If IsNumeric(ArrayGetIndex(allGroupNames39,memberValue)) Then IsGroup=t
If IsNumeric(ArrayGetIndex(allGroupNames40,memberValue)) Then IsGroup=t
If IsGroup Then
    c=UBound(foundGroups)
    If foundGroups(c)<>"" Then c=c+1
    ReDim Preserve foundGroups(c)
    foundGroups(c)=memberValue
End If
End Function
Function ProcessDenyAccessNames() As Boolean
    If Not debug Then On Error GoTo errHandle
    LogStep norm,CStr(Now)+tb+"ProcessDenyAccessNames" - begin|
    Dim daView As NotesView
    Dim daNote As NotesDocument
    Dim serverList As Variant
    Dim serverGroupList() As String
    Dim serverName As String
    Dim splitVar As Variant
    Dim groupVar As Variant
    Dim openSystemCheck As Boolean
    Dim x As Integer, y As Integer, z As Integer
    ReDim serverGroupList(0)
    serverList=GetAdminValue("ServerList") ' load the list of values from the list of Servers and their respective
    If serverList(0)="" Then GoTo exitThis
    Set daView=db.Getview("dagroups")
    Call daView.Refresh()
    daView.Autoupdate=f
    For x=0 To UBound(serverList)
        splitVar=Split(serverList(x),"|")
        If UBound(splitVar)>2 Then
            serverName=LCASE(splitVar(0))
            If splitVar(1)="Available" Then
                If splitVar(2)="" Then ' one of the servers does not use
                    openSystemCheck=t
                ElseIf splitVar(3)="" Or splitVar(4)="" Then ' one of the servers does not have secure HTTP/LDA
                    openSystemCheck=t
                Else
                    ReDim Preserve serverGroupList(z)
                    serverGroupList(z)=serverName
                    z=z+1
                End If
            End If
        End If
    Next
    SetAdminValue "AvalableServerCount",UBound(serverGroupList)
    ' sort the array so that we are comparing similar arrays
    RemoveDuplicates serverGroupList
    ' if this is an "Open System" (i.e. at least one server either does not use Deny Access Groups or does not h
    If openSystemCheck Then GoTo openSystem
    ' now cycle through each user in the "users" view to get a @DBLookup return result form the "Members" vie
    Dim userName As NotesName
    Dim userNote As NotesDocument

```

```

Dim userView As NotesView
Dim userNav As NotesViewNavigator
Dim userEntry As NotesViewEntry
Dim lookupView As NotesView
Dim lookupList As Variant
Dim lookupListNormalized As Variant
Dim formula As String
Dim daUser As Boolean
Dim canAccess() As String
Dim cantAccess() As String
Dim item As NotesItem
Set userView=db.Getview("users")
Call userView.Refresh()
userView.Autoupdate=f
Set lookupView=db.Getview("members")
Call lookupView.Refresh()
lookupView.Autoupdate=f
Set userNav=userView.Createviewnav()
Set userEntry=userNav.Getfirst()
While Not userEntry Is Nothing
    ReDim canAccess(0)
    ReDim cantAccess(0)
    daUser=t
    Set userName=New NotesName(userEntry.Columnvalues(0))
    formula=|list:=@DbLookup("": "NoCache";@DbName;"members";"+LCase(userName.Abbreviated)+|";
    lookupList=Evaluate(formula)
    Set userNote=userEntry.Document
    If lookupList(0)="" Then
        ' user was not found in any DA Groups. Make sure that they are not already listed as a DA User tho
        If userEntry.Columnvalues(1)="Yes" Then userNote.DAUser=""
        userNote.ServersNoAccess=""
        userNote.ServersNoAccessCount=0
        Call userNote.Removeitem("ServersAccessible")
        Call userNote.Replaceitemvalue("ServersAccessible",serverGroupList)
        userNote.ServersAccessibleCount=UBound(serverGroupList)+1
        Call userNote.Removeitem("ServersAvailable")
        Call userNote.Replaceitemvalue("ServersAvailable",serverGroupList)
        userNote.ServersAvailableCount=UBound(serverGroupList)+1
        Call userNote.Save(t,t,t)
    Else
        RemoveDuplicates lookupList
        y=0
        z=0
        daUser=t
        For x=0 To UBound(serverGroupList)
            If IsNumeric(ArrayGetIndex(lookupList,serverGroupList(x))) Then
                ReDim Preserve cantAccess(y)
                cantAccess(y)=serverGroupList(x)
                y=y+1
            Else
                ReDim Preserve canAccess(z)
                canAccess(z)=serverGroupList(x)
                z=z+1
                daUser=f
            End If
        Next
    End While
Next

```

```

        Set userNote=userEntry.Document
        If Not dauser Then ' user was not found in any DA Groups. Make sure that they are not already listed
            userNote.DAUser=""
        Else
            userNote.DAUser="Yes"
        End If
        Call userNote.Removeitem("ServersNoAccess")
        Call userNote.Replaceitemvalue("ServersNoAccess",cantAccess)
        If cantAccess(0)<>"" Then userNote.ServersNoAccessCount=UBound(cantAccess)+1 Else userNote.ServersNoAccessCount=0
        Call userNote.Removeitem("ServersAccessible")
        Call userNote.Replaceitemvalue("ServersAccessible",canAccess)
        If canAccess(0)<>"" Then userNote.ServersAccessibleCount=UBound(canAccess)+1 Else userNote.ServersAccessibleCount=0
        Call userNote.Removeitem("ServersAvailable")
        Call userNote.Replaceitemvalue("ServersAvailable",serverGroupList)
        userNote.ServersAvailableCount=UBound(serverGroupList)+1
        If UBound(serverGroupList)=UBound(cantAccess) Then
            userNote.PartialDAUser=""
            userNote.DAUser="true"
        ElseIf UBound(serverGroupList)<>UBound(canAccess) Then
            userNote.PartialDAUser="true"
        Else
            userNote.PartialDAUser=""
        End If
        Call userNote.Save(t,t,t)
    End If
    Set userEntry=userNav.Getnext(userEntry)
Wend
' get the count of Deny Access users
Set userView=db.Getview("dauser")
Call userView.Refresh()
userView.Autoupdate=f
Set userNav=userView.Createviewnav()
SetAdminValue "DAUserCount",userNav.Count
' get the count of Authorized Users
Set userView=db.Getview("valid")
Call userView.Refresh()
userView.Autoupdate=f
Set userNav=userView.Createviewnav()
SetAdminValue "ValidUserCount",userNav.Count
' get the count of Mail Users
Set userView=db.Getview("mailuser")
Call userView.Refresh()
userView.Autoupdate=f
Set userNav=userView.Createviewnav()
SetAdminValue "MailUserCount",userNav.Count
exitThis:
    LogStep norm,CStr(Now)+tb+|"ProcessDenyAccessNames" - finished|
    ProcessDenyAccessNames=t
errHandle:
    If Err<>0 Then
        errString=Error
        errNum=Err
        errLine=Erl
        ReportError "There was an Error during the (ProcessDenyAccessNames) function. The details of the Error are: " & errString & " Error Number: " & errNum & " Error Line: " & errLine
    End If
Exit Function

```

openSystem:

```
' there is one or more domino servers that either do not have a DA Group on the server doc or the DA Group
Dim userDC As NotesDocumentCollection
Set userView=db.Getview("invalid")
Call userView.Refresh()
userView.Autoupdate=f
Set userDC=userView.Getalldocumentsbykey("all")
If userDC.Count<>0 Then Call userDC.Stampall("DAUser","")
' set the count of Deny Access users to zero
SetAdminValue "DAUserCount",0
' get the count of Mail Users
Set userView=db.Getview("mailuser")
Call userView.Refresh()
userView.Autoupdate=f
Set userNav=userView.Createviewnav()
SetAdminValue "MailUserCount",userNav.Count
' get the count of Authorized Users
Set userView=db.Getview("valid")
Call userView.Refresh()
userView.Autoupdate=f
Set userDC=userView.Getalldocumentsbykey("1")
If userDC.Count<>0 Then
    Call userDC.Stampall("ServersAccessible",serverGroupList)
    Call userDC.Stampall("ServersAvailable",serverGroupList)
End If
Set userNav=userView.Createviewnav()
SetAdminValue "ValidUserCount",userNav.Count
LogStep norm,CStr(Now)+tb+"ProcessDenyAccessNames" - finished|
ProcessDenyAccessNames=t
Exit Function
```

End Function

Sub SelectNAB(adminServer As String)

```
Dim svr As NotesName
Dim newList As Variant
Dim newValue As String
Dim dbFile As Variant
Dim sourceInfo(1) As String
Dim selectedDB As NotesDatabase
Dim x As Integer
MsgBox "Your HCL Domino Administration server could not be automatically located. You will now be asked to select a directory."
```

selectNAB:

```
If ws Is Nothing Then Set ws=New NotesUIWorkspace
newList=ws.Prompt(13,"Select HCL Domino Directory","Select the HCL Domino Directory on your HCL Domino server")
If IsEmpty(newList) Then Exit Sub
If UBound(newList)>0 Then ' let's see if they selected multiple dbs
    If UBound(newList)<>2 Then GoTo tryAgain
    If InStr(1,newList(1),"~")<>0 Then GoTo tryAgain
    If InStr(1,newList(1),".nsf" "|,5)<>0 Then GoTo tryAgain
End If
If UBound(newList)=2 Then ' array - extract proper values
    Set svr=New NotesName(newList(0))
    If svr.Common="" Then
        MsgBox "You selected a local HCL Domino directory. Please select the HCL Domino Directory on your HCL Domino server."
        GoTo selectNAB
    End If
    sourceInfo(0)=svr.Abbreviated
```

```

        sourceInfo(1)=newList(1)
Else
    ' single value, split it up
    newValue=StrToken(newList(0),"|",2)
    Set svr=New NotesName(StrToken(newValue,"~",1))
    If svr.Common="" Then
        MsgBox "You selected a local HCL Domino directory. Please select the HCL Domino Directory on
        GoTo selectNAB
    End If
    sourceInfo(0)=svr.Abbreviated
    sourceInfo(1)=StrToken(newValue,"~",2)
End If
If InStr(1,sourceInfo(1),"\")<>0 Then
    dbFile=Split(Left(sourceInfo(1),Len(sourceInfo(1))-4),"\")
    dbFile=dbFile(UBound(dbFile))
Else
    dbFile=Left(sourceInfo(1),Len(sourceInfo(1))-4)
End If
If sourceInfo(0)="Local Workstation" Then Set selectedDB=session.GetDatabase("",sourceInfo(1)) Else Se
If Not selectedDB.IsOpen Then Call selectedDB.Open("", "")
If Not selectedDB.IsOpen Then MsgBox "The HCL Domino Directory you selected is not accessible. Pleas
Dim checkForm As NotesForm
Dim checkView As NotesView
Set checkForm=selectedDB.Getform("Person")
If checkForm Is Nothing Then GoTo notValid
Set checkForm=selectedDB.Getform("Server")
If checkForm Is Nothing Then GoTo notValid
Set checkView=selectedDB.Getview("($VIMGroups)")
If checkView Is Nothing Then GoTo notValid
Set checkView=selectedDB.Getview("($VIMPeople)")
If checkView Is Nothing Then GoTo notValid
Set checkView=selectedDB.Getview("($Servers)")
If checkView Is Nothing Then GoTo notValid
adminServer=svr.Abbreviated
SetAdminValue "AdminServer",svr.Abbreviated
nabFilename=sourceInfo(1)
Exit Sub
notValid:
    If MsgBox("The HCL Notes application that you chose is not a valid HCL Domino Directory,"+cr+cr+"Do yo
    Exit Sub
tryagain:
    MsgBox "Only one HCL Domino Directory can be selected. Please try again."
    GoTo selectNAB
End Sub
Function GetPersonDocuments(processNAB As NotesDatabase) As Boolean
    If Not debug Then On Error GoTo errHandle
    LogStep norm,CStr(Now)+tb+|"GetPersonDocuments" - begin|
    LogStep norm,CStr(Now)+tb+|Scanning Person documents in the Domino directory -"|+processNAB.Title+
    Dim nabNote As NotesDocument
    Dim nabView As NotesView
    Dim userNote As NotesDocument
    Dim userView As NotesView
    Dim viewNAV As NotesViewNavigator
    Dim externalNABName As String
    Dim externalNAB As Boolean
    Dim externalORGs As Variant
    ReDim externalORGs(0)

```



```

externalORGs(0)=""
Dim tmpVal As Variant
Dim splitVar As Variant
Dim x As Integer
Dim count As Long
Dim navCount As Long
Dim logTXT As String
Dim key As String
Dim update As Boolean
Dim certExpiryDate As NotesDateTime
Dim nowDate As New NotesDateTime(Now)
Dim certAvail As Boolean
Dim pwAvail As Boolean
Dim serverList As Variant
Dim httpSecure As String
Dim ldapSecure As String
httpSecure=GetAdminValue("HTTPSecure")(0)
ldapSecure=GetAdminValue("LDAPSecure")(0)
' if an external directory was chosen to identify CCX users, check to see if the current directory is the one th
If GetAdminValue("DAListExternal")(0)<>"" Then
    tmpVal=GetAdminValue("DAListExternal")
    tmpVal=Split(tmpVal(0),"!!")
    externalNABName=Left(tmpVal(1),Len(tmpVal(1))-1)
    If LCase(processNAB.Filename)=externalNABName Then externalNAB=t
End If
' if ORG values were chosen to identify CCX users, get the list of ORGs selected
If GetAdminValue("OrgsExternal")(0)<>"" Then externalORGs=GetAdminValue("OrgsExternal")
Set nabView=processNAB.Getview("($VIMPeople)")
Call nabView.Refresh()
nabView.Autoupdate=f
Set viewNAV=nabView.Createviewnav()
navCount=viewNAV.Count
LogStep norm,CStr(Now)+tb+|# of Person documents in Domino Directory = |+CStr(navCount)
Set userView=db.Getview("users")
Call userView.Refresh()
userView.Autoupdate=f
Set nabNote=nabView.Getfirstdocument()
While Not nabNote Is Nothing
    certAvail=f
    pwAvail=f
    If nabNote.isvalid And Not nabNote.HasItem("$REF") Then
        update=f
        key=LCase(nabNote.FullName(0))
        Set userNote=userView.Getdocumentbykey(key,t)
        If userNote Is Nothing Then
            Set userNote=db.Createdocument()
            userNote.Form="User"
            update=t
        End If
        If Not update Then If userNote.NABNoteUpdated(0)<>nabNote.Lastmodified Then update=t
        If nabNote.HasItem("Certificate") Then If nabNote.Certificate(0)<>"" Then certAvail=t
        If nabNote.HasItem("HTTPPassword") Then If nabNote.HTTPPassword(0)<>"" Then pwAvail=t
        If Not certAvail And Not pwAvail Then userNote.UserType="Unknown / Unlicensed"
        If Not userNote.hasitem("UserType") Then
            If nabNote.Hasitem("Certificate") Then
                If nabNote.Certificate(0)<>"" Then

```

```

        userNote.UserType="Client"
        tmpVal=Evaluate(|@Date(@Certificate([Expiration];Certificate))|,nabNote)
        If IsArray(tmpVal) Then
            Set certExpiryDate=New NotesDateTime(tmpVal(0))
            userNote.NABNoteCertExpiry=certExpiryDate.Lslocaltime
            If certExpiryDate.Timedifference(nowDate)<0 Then
                userNote.CertificateExpired="true"
                update=t
            Else
                If userNote.Hasitem("CertificateExpired") Then
                    Call userNote.Removeitem("CertificateExpired")
                    update=t
                End If
            End If
        End If
    End If
End If
If Not userNote.hasitem("UserType") And nabNote.Hasitem("HTTPPassword") Then
    If userNote.HTTPPassword(0)<>"" Then
        userNote.UserType="CAL"
        update=t
    End If
End If
If Not userNote.Hasitem("UserType") Then
    ' the Person document does not have a Notes Certificate or an Internet Password. this is most
    userNote.UserType="Unknown / Unlicensed"
    update=t
End If
If update Then
    userNote.NABNoteUpdated=nabNote.Lastmodified
    userNote.NABNoteUNID=nabNote.Universalid
    userNote.NABServer=processNAB.Server
    userNote.NABPath=processNAB.Filepath
    userNote.NABName=processNAB.Title
    If nabNote.MailSystem(0)="1" Then
        userNote.MailUser="true"
        userNote.NABNoteMailServer=nabNote.MailServer(0)
        userNote.NABNoteMailFile=nabNote.MailFile(0)
    Else
        Call userNote.Removeitem("MailUser")
        Call userNote.Removeitem("MailServer")
        Call userNote.Removeitem("MailFile")
    End If
    userNote.FullName=nabNote.FullName(0)
    userNote.InternetAddress=nabNote.InternetAddress(0)
    If nabNote.HTTPPassword(0)<>"" Then userNote.HTTPPasswordSet="true" Else userNote.HTTPPasswordSet="false"
    userNote.HTTPSecure=httpSecure
    userNote.LDAPSecure=ldapSecure
    If externalNAB Then userNote.ExternalNAB="true"
    If externalORGs(0)<>"" Then
        For x=0 To UBound(externalORGs)
            If InStr(1,userNote.FullName(0),externalORGs(x),5)<>0 Then userNote.ExternalORGs(x)=externalORGs(x)
        Next
    End If
    If deepLog Then

```

```

        count=count+1
        logTXT=CStr(count)+" of "+CStr(navCount)+" - Person = "+userNote.FullName(0)+" (" +userNote
        LogStep norm,CStr(Now)+tb+logTXT
    End If
    Call userNote.Save(t,t,t)
End If
End If
Set nabNote=nabView.Getnextdocument(nabNote)
Wend
LogStep norm,CStr(Now)+tb+"|GetPersonDocuments" - finished|
GetPersonDocuments=t
errHandle:
If Err<>0 Then
    errString=Error
    errNum=Err
    errLine=Erl
    ReportError "There was an Error during the (GetPersonDocuments) function. The details of the Error are: " & errString
End If
Exit Function
End Function

```

Name:

Last Modification:

slUtilities

01/11/2023 02:55:16 PM

LotusScript Code:

```

Option Public
Option Declare
Dim ws As NotesUIWorkspace
Dim ui As NotesUIDocument
Dim note As NotesDocument
Dim session As NotesSession
Dim db As NotesDatabase
Dim view As NotesView
Dim admin As NotesDocument
Dim currentUser As NotesName
Dim mainO As String
Dim cr As String
Dim tb As String
Const t=True
Const f=False
Dim errString As String
Dim errLine As Integer
Dim errNum As Integer
Dim errCritical As Boolean
Const debug=f
Const deepLog=f
'Dim debug As Boolean
'Dim deepLog As Boolean
Const info=1
Const sec=2
Const warn=3
Const norm=0
Sub Initialize
%rem
'    Dim openSession As New NotesSession
'    Dim enviro As String
'    Set session=New NotesSession

```

```

        enviro=session.Getenvironmentstring("DLAU_Debug",True)
        If enviro<>"" Then debug=t
        enviro=session.Getenvironmentstring("DLAU_Logging",True)
        If enviro<>"" Then deepLog=t
    ,    Set openSession=Nothing
%end rem
End Sub
Function GetAdminValue(fldName As String) As Variant
    Dim empty(0) As String
    If admin Is Nothing Then
        Set view=db.Getview("config")
        Set admin=view.Getdocumentbykey("config",t)
        If admin Is Nothing Then
            Set admin=db.Createdocument()
            admin.Form="config"
            Call admin.Save(t,t,t)
        End If
    End If
    GetAdminValue=empty
    If admin.Hasitem(fldName) Then GetAdminValue=admin.Getfirstitem(fldName).Values
End Function
Sub SetAdminValue (fldName As String,value As Variant)
    Dim item As NotesItem
    Dim summaryFields(8) As String
    summaryFields(0)="adminserver"
    summaryFields(1)="observationswarning"
    summaryFields(2)="observationssecurity"
    summaryFields(3)="observationsinformation"
    summaryFields(4)="validusercount"
    summaryFields(5)="persondoccount"
    summaryFields(6)="groupdoccount"
    summaryFields(7)="dausercount"
    summaryFields(8)="form"
    If admin Is Nothing Then
        Set view=db.Getview("config")
        Set admin=view.Getdocumentbykey("config",t)
        If admin Is Nothing Then
            Set admin=db.Createdocument()
            admin.Form="config"
        End If
    End If
    Call admin.Removeitem(fldName)
    Call admin.Replaceitemvalue(fldName,value)
    If Not IsNumeric(ArrayGetIndex(summaryFields,LCase(fldName))) Then
        Set item=admin.Getfirstitem(fldName)
        item.Issummary=f
    End If
    Call admin.Save(t,t,t)
End Sub
Sub LogStep (color As Integer,logValue As String)
    Dim rtltem As NotesRichTextItem
    Dim rtStyle As NotesRichTextStyle
    If session Is Nothing Then QO
    If admin Is Nothing Then
        Set view=db.Getview("config")
        Set admin=view.Getdocumentbykey("config",t)

```

```

        If admin Is Nothing Then
            Set admin=db.Createdocument()
            admin.Form="config"
            Set rtltem=New NotesRichTextItem(admin,"Debug")
        Else
            If admin.hasitem("Debug") Then
                Set rtltem=admin.GetFirstItem("Debug")
            Else
                Set rtltem=New NotesRichTextItem(admin,"Debug")
            End If
        End If
    Else
        If admin.hasitem("Debug") Then
            Set rtltem=admin.GetFirstItem("Debug")
        Else
            Set rtltem=New NotesRichTextItem(admin,"Debug")
        End If
    End If
    Set rtStyle=session.CreateRichTextStyle
    If color=warn Then
        rtStyle.Bold=t
        rtStyle.NotesColor=COLOR_MAGENTA
    ElseIf color=info Then
        rtStyle.Bold=f
        rtStyle.NotesColor=COLOR_BLUE
    ElseIf color=sec Then
        rtStyle.Bold=t
        rtStyle.NotesColor=COLOR_RED
    Else
        rtStyle.Bold=f
        rtStyle.NotesColor=COLOR_BLACK
    End If
    Call rtltem.AppendStyle(rtStyle)
    Call rtltem.appendtext(logValue)
    Call rtltem.addnewline(1,t)
    Call admin.Save(t,t,t)
End Sub
Sub AppendAdminValue (fldName As String, value As Variant)
    Dim adminVar As Variant
    Dim upper As Integer
    adminVar=GetAdminValue(fldName)
    If adminVar(0)="" Then
        Call admin.Removeitem(fldName)
        Call admin.Replaceitemvalue(fldName,value)
    Else
        upper=UBound(adminVar)+1
        ReDim Preserve adminVar(upper)
        adminVar(upper)=value
        Call admin.Removeitem(fldName)
        Call admin.Replaceitemvalue(fldName,adminVar)
    End If
    Call admin.Save(t,t,t)
End Sub
Function DifferenceBetweenTwoArrays (array1 As Variant, array2 As Variant) As Variant
    Dim c&, foundValue As Variant
    Dim differenceArray() As String

```

```

ReDim differenceArray(0)
c& = 0
ReDim differenceArray(0)
ForAll arr1 In array1
    foundValue = False
    ForAll arr2 In array2
        If LCase(arr2) = LCase(arr1) Then
            foundValue = True
        End If
    End ForAll
    If foundValue = False Then
        ReDim Preserve differenceArray(c&)
        differenceArray(c&) = arr1
        c& = c& + 1
    End If
End ForAll
DifferenceBetweenTwoArrays=differenceArray
End Function
Sub RemoveDuplicates (values As Variant)
    Call SortArray (values)
    Dim finalvalues () As String
    ReDim finalvalues (0)
    If UBound (values) = 0 Then Exit Sub
    Dim x As Long
    While x < UBound (values)
        Do While (values (x) = values (x + 1)) Or (values (x) = "")
            'value at x+1 is the same as value at x, skip it (don't add it final array thereby removing the duplicate)
            x = x + 1
            If x = UBound (values) Then Exit Do
        Loop
        If values (x) <> "" Then
            If UBound (finalvalues) = 0 And finalvalues (0) = "" Then
                ' first element
                finalvalues (0) = values (x)
            Else
                ReDim Preserve finalvalues (UBound (finalvalues) + 1)
                finalvalues (UBound (finalvalues)) = values (x)
            End If
        End If
        x = x + 1
    Wend
    ' last element, add if different from last element in finalvalues, skip otherwise (duplicate)
    If values (UBound (values)) <> finalvalues (UBound (finalvalues)) Then
        ReDim Preserve finalvalues (UBound (finalvalues) + 1)
        finalvalues (UBound (finalvalues)) = values (UBound (values))
    End If
    ' repopulate input parameter values with finalvalues
    ReDim values (UBound (finalvalues)) As String
    Dim count As Long
    ForAll a In finalvalues
        values (count) = a
        count = count + 1
    End ForAll
End Sub
Function GetProfileValue(profileName As String, perName As String, fldName As String) As Variant
    Dim profile As NotesDocument

```

```

    If perName="" Then
        Set profile=db.Getprofiledocument(profileName)
    Else
        Set profile=db.Getprofiledocument(profileName,perName)
    End If
    If profile.Hasitem(fldName) Then
        GetProfileValue=profile.Getfirstitem(fldName).Values
    Else
        Dim empty(0) As String
        GetProfileValue=empty
    End If
End Function
Sub SortArray (A As Variant)
    Dim gap As Integer
    Dim low As Integer
    Dim high As Integer
    Dim i As Integer
    Dim makeswap As Variant
    Dim PushPop As Variant
    low = LBound (A)
    high = UBound (A)
    gap = Int (high / 2)
    While gap >= 1
        Do
            makeswap = False
            For i = low To (high - gap)
                If LCase ( A (i) ) > LCase ( A ( i + gap ) ) Then
                    PushPop = A (i)
                    A (i) = A (i + gap)
                    A (i + gap) = PushPop
                    makeswap = True
                End If
            Next i
            Loop Until Not makeswap
            gap = Int (gap / 2)
        Wend
    End Sub
Sub SetProfileValue(profileName As String, perName As String, fldName As String, value As Variant)
    Dim profile As NotesDocument
    If perName="" Then
        Set profile=db.Getprofiledocument(profileName)
    Else
        Set profile=db.Getprofiledocument(profileName,perName)
    End If
    Call profile.Removeitem(fldName)
    Call profile.Replaceitemvalue(fldName,value)
    Call profile.Save(t,t,t)
End Sub
Sub Observation(observeNum As Integer,observeMessage As String, svrName As String)
    Dim currentValues As Variant
    Dim count As Integer
    Dim newValue As String
    Dim warn As Integer
    Dim security As Integer
    Dim info As Integer
    Dim adminVal As String

```

```

Dim increaseCount As Boolean
ReDim currentValues(0)
newValue=observeMessage+"|"+CStr(observeNum)+"~"+observeMessage
If svrName<>"" Then newValue=newValue+"~"+svrName
currentValues=GetAdminValue("Observations")
If currentValues(0)="" Then
    currentValues(0)=newValue
    increaseCount=t
Else
    If Not IsNumeric(ArrayGetIndex(currentValues,newValue)) Then
        count=UBound(currentValues)+1
        ReDim Preserve currentValues(count)
        currentValues(count)=newValue
        increaseCount=t
    End If
End If
If increaseCount Then
    ' get the counts of each type of observation
    adminVal=GetAdminValue("ObservationsWarning")(0)
    If adminVal<>"" Then warn=CInt(adminVal)
    adminVal=GetAdminValue("ObservationsSecurity")(0)
    If adminVal<>"" Then security=CInt(adminVal)
    adminVal=GetAdminValue("ObservationsInformation")(0)
    If adminVal<>"" Then info=CInt(adminVal)
    Select Case observeNum
        Case 1001,1008 : warn=warn+1
        Case 1002,1003,1005,1006,1009 : security=security+1
        Case 1004,1007 : info=info+1
    End Select
    ' set the values onto the config document
    SetAdminValue "Observations",currentValues
    SetAdminValue "ObservationsWarning",warn
    SetAdminValue "ObservationsSecurity",security
    SetAdminValue "ObservationsInformation",info
End If
End Sub
Sub ReportError (errMessage As String)
    If errString="" Then
        errString=Error
        errNum=Err
        errLine=Erl
    End If
    LogStep warn,errMessage+cr+cr+"Error: "+errString+" (#: "+CStr(errNum)+", line #: "+CStr(errLine)+")"
    MsgBox errMessage+cr+cr+"Error: "+errString+cr+"#: "+CStr(errNum)+cr+cr+"line #: "+CStr(errLine)
    Print errMessage+cr+cr+"Error: "+errString+cr+"#: "+CStr(errNum)+", line #: "+CStr(errLine)
    SetAdminValue "Critical","true"
    errString=""
    errNum=0
    errLine=0
End Sub
Sub QO
    If Not debug Then On Error GoTo errHandle
    Set session=New NotesSession
    Set db=session.Currentdatabase
    Set currentUser=New NotesName(session.Username)
    mainO=currentUser.Organization

```



```

errHandle:
  If Err<>0 Then
    errString=Error
    errNum=Err
    errLine=Erl
    ReportError "There was an Error opening this HCL Notes application in the (QO) event. The details of t
  End If
  Exit Sub
End Sub
Sub PO
  If Not debug Then On Error GoTo errHandle
  Set ws=New NotesUIWorkspace
  Set ui=ws.Currentdocument
  Set note=ui.Document
errHandle:
  If Err<>0 Then
    errString=Error
    errNum=Err
    errLine=Erl
    ReportError "There was an Error opening this HCL Notes application in the (PO) event. The details of t
  End If
  Exit Sub
End Sub

```

Agent Information

Name:

Alias:
 Last Modification:
 Comment:
 Shared Agent:
 Type:
 State:
 Trigger:
 Acts On:

(MakeHistory)

MakeHistory
 01/11/2023 02:55:17 PM
 [Not Assigned]
 Yes
 LotusScript
 Enabled
 Manually From Actions Menu
 None

LotusScript Code:

```

Option Public
Option Declare
Sub Initialize
  Const t=True
  Dim session As New NotesSession
  Dim db As NotesDatabase
  Dim view As NotesView
  Dim doc As NotesDocument
  Dim change As NotesDocument
  Dim dc As NotesDocumentCollection
  Set db=session.Currentdatabase
  Set view=db.Getview("config")
  Set doc=view.Getfirstdocument()
  If Not doc Is Nothing Then
    If Not doc.Hasitem("AllScansComplete") Then Exit Sub
    doc.Form="History"
    Call doc.Save(t,t,t)
  End If
  Set view=db.Getview("dagroups")

```

```

Set doc=view.Getfirstdocument()
While Not doc Is Nothing
    Set change=doc
    Set doc=view.Getnextdocument(doc)
    change.Form="GroupHistory"
    Call change.Save(t,t,t)
Wend
Set dc=db.Search(|Form="User"|, Nothing,0)
If dc.Count<>0 Then
    Call dc.Stampall("Form", "OldUsers")
    Call dc.Removeall(t)
End If
End Sub

```

Name:

Alias:
 Last Modification:
 Comment:
 Shared Agent:
 Type:
 State:
 Trigger:
 Acts On:

(PullInfoConfig)

PullInfoConfig
 01/11/2023 02:55:14 PM
 [Not Assigned]
 Yes
 LotusScript
 Enabled
 Manually From Actions Menu
 None

LotusScript Code:

```

Option Public
Option Declare
Use "slCore"
Sub Initialize
    QO
    SetAdminValue "ConfigPullStatus","Running"
    LogStep norm,CStr(Now)+tb+|In "(PullInfoConfig)" agent|
    LogStep norm,CStr(Now)+tb+|Starting "GetAdminServerInfo" function|
    GetAdminServerInfo
    LogStep norm,CStr(Now)+tb+|Finished "GetAdminServerInfo" function|
    LogStep norm,CStr(Now)+tb+|Starting "CheckServerSettings" function|
    CheckServerSettings
    LogStep norm,CStr(Now)+tb+|Finished "CheckServerSettings" function|
    SetAdminValue "ConfigPullStatus","Complete"
    LogStep norm,CStr(Now)+tb+|"(PullInfoConfig)" agent finished|
End Sub

```

Name:

Alias:
 Last Modification:
 Comment:
 Shared Agent:
 Type:
 State:
 Trigger:
 Acts On:

(PullInfoDAGroups)

PullInfoDAGroups
 01/11/2023 02:55:14 PM
 [Not Assigned]
 Yes
 LotusScript
 Enabled
 Manually From Actions Menu
 None

LotusScript Code:

```

Option Public
Option Declare
Use "slCore"
Sub Initialize
    QO

```

```

LogStep norm,CStr(Now)+tb+|In "(PullInfoDAGroups)" agent|
Dim nabSvr As String
nabSvr=GetAdminValue("AdminServer")(0)
LogStep norm,CStr(Now)+tb+|Obtaining Domino Directory|
Set nab=session.Getdatabase(nabSvr,"names.nsf")
If Not nab.isopen Then Call nab.Open("", "")
If Not nab.Isopen Then
    SetAdminValue "DADocSyncError","Could not scan the Deny Access Grooup documents - Domino Dir
    SetAdminValue "DADocSyncStatus","Complete"
    LogStep warn,CStr(Now)+tb+"Could not scan the Deny Access Grooup documents - Domino Directory
    Exit Sub
End If
SetAdminValue "DADocSyncError", ""
LogStep norm,CStr(Now)+tb+|Obtaining Deny Access Groups|
SetAdminValue "DADocSyncStatus","Synching"
If Not GetDenyAccessGroups Then
    SetAdminValue "DADocSyncError","Could not scan the Deny Access Group documents - error in sync
    SetAdminValue "DADocSyncStatus","Complete"
    LogStep warn,CStr(Now)+tb+"Could not scan the Deny Access Group documents - error in synch routi
    Exit Sub
End If
SetAdminValue "DADocSyncdate",Today
SetAdminValue "DADocSyncStatus","Complete"
LogStep norm,CStr(Now)+tb+|"(PullInfoDAGroups)" agent finished|
End Sub

```

Name:

Alias:

Last Modification:

Comment:

Shared Agent:

Type:

State:

Trigger:

Acts On:

(PullInfoEntitlement)

PullInfoEntitlement

01/11/2023 02:55:15 PM

[Not Assigned]

Yes

LotusScript

Enabled

Manually From Actions Menu

None

LotusScript Code:

Option Public

Option Declare

Use "slCore"

Sub Initialize

QO

LogStep norm,CStr(Now)+tb+|In "(PullInfoEntitlement)" agent|

Dim splitVar As Variant

Dim checkDB As NotesDatabase

Dim entArray As Variant

Dim daVal As String

Dim x As Integer

ReDim entArray(0)

entArray(0)=""

SetAdminValue "EntitlementSyncStatus","Starting"

LogStep norm,CStr(Now)+tb+|Retrieving Entitlement Information|

entArray=GetAdminValue("EntitlementTrackingServers")

daVal=GetAdminValue("UserReportLocation")(0)

If entArray(0)<>"" Then

LogStep norm,CStr(Now)+tb+|Scanning Entitlement system|

Set checkDB=session.Getdatabase(GetAdminValue("AdminServer")(0), "entitlements.nsf")

```

        If Not checkDB.isopen Then Call checkDB.Open("", "")
        If checkDB.isopen Then If Not ProcessEntitlements(checkDB,f) Then Exit Sub
    ElseIf daVal<>"" Then
        LogStep norm,CStr(Now)+tb+|Scanning User Report Results|
        splitVar=Split(daVal,"|")
        splitVar=Split(splitVar(1),"~")
        If splitVar(0)="Local Workstation" Then Set checkDB=session.Getdatabase("",splitVar(1)) Else Set che
        If Not checkDB.isopen Then Call checkDB.Open("", "")
        If checkDB.isopen Then If Not ProcessEntitlements(checkDB,t) Then Exit Sub
    End If
    SetAdminValue "EntitlementSyncdate",Today
    SetAdminValue "EntitlementSyncStatus","Complete"
    LogStep norm,CStr(Now)+tb+|"(PullInfoEntitlement)" agent finished|
End Sub

```

Name:

Alias:
 Last Modification:
 Comment:
 Shared Agent:
 Type:
 State:
 Trigger:
 Acts On:

(PullInfoUser)

PullInfoUser
 01/11/2023 02:55:14 PM
 [Not Assigned]
 Yes
 LotusScript
 Enabled
 Manually From Actions Menu
 None

LotusScript Code:

```

Option Public
Option Declare
Use "slCore"
Sub Initialize
    QO
    LogStep norm,CStr(Now)+tb+|In "(PullInfoUser)" agent|
    Dim tmpVal As Variant
    Dim splitVar As Variant
    Dim checkDB As NotesDatabase
    Dim x As Integer
    Dim userCount As Long
    SetAdminValue "NABPersonDocSyncStatus","Starting"
    LogStep norm,CStr(Now)+tb+|Obtaining Domino Directory|
    Dim nabSvr As String
    nabSvr=GetAdminValue("AdminServer")(0)
    Set nab=session.Getdatabase(nabSvr,"names.nsf")
    If Not nab.isopen Then Call nab.Open("", "")
    If Not nab.Isopen Then
        SetAdminValue "NABPersonDocSyncError","Could not scan the Person documents - Domino Directory"
        Exit Sub
    End If
    SetAdminValue "NABPersonDocSyncError",""
    LogStep norm,CStr(Now)+tb+|Obtaining Person document information|
    ' get Person documents from the main Admin Domino Directory
    SetAdminValue "NABPersonDocSyncStatus","Synching"
    If Not GetPersonDocuments(nab) Then
        SetAdminValue "NABPersonDocSyncError","Could not scan the Person documents - error in synch rou
        SetAdminValue "NABPersonDocSyncStatus","Complete"
        Exit Sub
    End If
    tmpVal=GetAdminValue("DAList")

```

```

If tmpVal(0)<>"" Then
    For x=0 To UBound(tmpVal)
        LogStep norm,CStr(Now)+tb+|Scanning additional Domino Directories|
        splitVar=Split(tmpVal(x),"~")
        splitVar=Split(splitVar(1),"|")
        Set checkDB=session.Getdatabase(splitVar(0),splitVar(1))
        If Not checkDB.isopen Then Call checkDB.Open("", "")
        If checkDB.Isopen Then
            If Not GetPersonDocuments(checkDB) Then
                SetAdminValue "NABPersonDocSyncError","Could not scan the Person documents in add
                SetAdminValue "NABPersonDocSyncStatus","Complete"
                Exit Sub
            End If
        End If
    Next
End If
' set how many Person documents were found
Dim userView As NotesView
Set userView=db.Getview("users")
Call userView.Refresh()
userView.Autoupdate=f
Dim userNav As NotesViewNavigator
Set userNav=userView.Createviewnav()
SetAdminValue "PersonDocCount",userNav.Count
' get License Usage documents from any user License Tracking dbs that were discovered
userCount=0
tmpVal=GetAdminValue("LicenseTrackingServers")
If tmpVal(0)<>"" Then
    LogStep norm,CStr(Now)+tb+|Scanning License Tracking system|
    For x=0 To UBound(tmpVal)
        splitVar=Split(tmpVal(x),"|")
        Set checkDB=session.Getdatabase(splitVar(1),"userlicenses.nsf")
        If Not checkDB.isopen Then Call checkDB.Open("", "")
        If checkDB.Isopen Then
            If Not GetLicenseTrackingInfo(checkDB,userCount) Then
                SetAdminValue "NABPersonDocSyncError","Could not scan the User License Tracking inf
                SetAdminValue "NABPersonDocSyncStatus","Complete"
                Exit Sub
            End If
        End If
    Next
End If
SetAdminValue "NABPersonDocSyncdate",Today
SetAdminValue "NABPersonDocSyncStatus","Complete"
LogStep norm,CStr(Now)+tb+|"(PullInfoUser)" agent finished|
End Sub

```

Page Information

Name:

Last Modification:

Comment:

Treat contents as HTML:

Subcomponents:

pgBlank

01/11/2023 02:55:16 PM

[Not Assigned]

No

[None]
JavaScript & HTML Code:
[None]
LotusScript Code:
[None]

Name:

Alias:
Last Modification:
Comment:
Treat contents as HTML:

pgFooter

black
01/11/2023 02:55:13 PM
[Not Assigned]
No

Subcomponents:

[None]
JavaScript & HTML Code:
[None]
LotusScript Code:
[None]

Name:

Last Modification:
Comment:
Window Title Formula:
Treat contents as HTML:

pgHeader

01/11/2023 02:55:12 PM
[Not Assigned]
@DbTitle
No

Subcomponents:

[None]
JavaScript & HTML Code:
[None]
LotusScript Code:

(Globals)

```
Option Public
Option Declare
Use "slCore"
Dim openWS As NotesUIWorkspace
Dim openUI As NotesUIDocument
Dim openNote As NotesDocument
Dim openSession As NotesSession
Dim openDB As NotesDatabase
Dim openView As NotesView
Dim openAdmin As NotesDocument
Const showlicense=f
Function openGetAdminValue(fldName As String) As Variant
    On Error Goto errHandle
    Dim empty(0) As String
    If openAdmin Is Nothing Then
        If openDB Is Nothing Then Print "no handle on DB"
        Set openView=openDB.Getview("config")
        If openView Is Nothing Then Print "no handle on VIEW"
        Set openAdmin=openView.Getdocumentbykey("config",True)
        If openAdmin Is Nothing Then
            Set openAdmin=openDb.Createdocument()
            openAdmin.Form="config"
            Call openAdmin.Save(True,True,True)
```

```

        End If
    End If
    openGetAdminValue=empty
    If openAdmin.Hasitem(fldName) Then openGetAdminValue=openAdmin.Getfirstitem(fldName).Values
errHandle:
'   If Err<>0 Then MsgBox "ERROR in openGetAdminValue: "+Error+Chr(10)+"#: "+Cstr(Err)+"", line: "+Cstr(E
Exit Function
End Function
(Page) pgHeader
Sub Postopen(Source As Notesuidocument)
    Set openWS=New NotesUIWorkspace
    Set openUI=source
    Set openNote=source.Document
    Set openSession=New NotesSession
    Set openDB=openSession.CurrentDatabase
    If showlicense Then
        Dim dialog As New NotesDocument(openDB)
        If Not openWS.DialogBox("Agreement",True,True,True,False,False,False,"Terms and Conditions",dial
        If dialog.Accept(0)="2" Then Call openUI.Close
    End If
    If Not CheckDesign Then
        Call openWS.SetTargetFrame ("Inner")
        Call openWS.OpenPage("pgNotSigned")
    ElseIf Instr(1,openSession.Platform,"Windows",5)=0 Then
        Call openWS.SetTargetFrame ("Inner")
        Call openWS.OpenPage("pgNomad")
    ElseIf openGetAdminValue("ScansComplete")(0)<>"" Then
        Call openWS.SetTargetFrame ("Inner")
        Call openWS.OpenFrameSet("fsInner")
    Else
        Call openWS.SetTargetFrame ("Inner")
        Call openWS.OpenFrameSet("fsLaunch")
    End If
End Sub

```

Name:

Last Modification:

Comment:

Treat contents as HTML:

Subcomponents:

[None]

JavaScript & HTML Code:

[None]

LotusScript Code:

[None]

pgNomad

01/11/2023 02:55:16 PM

[Not Assigned]

No

Name:

Last Modification:

Comment:

Treat contents as HTML:

Subcomponents:

Hotspot

Type:

pgNotSigned

01/11/2023 02:55:19 PM

[Not Assigned]

No

Action Hotspot

Caption:	Re-sign Design
Is Scripted:	Yes, see below
JavaScript & HTML Code:	
[None]	
LotusScript Code:	
(Hotspot) Re-sign Design	
Sub Click(Source As Button) Dim ws As New NotesUIWorkspace Dim session As New NotesSession Dim db As NotesDatabase Dim view As NotesView Dim viewNote As NotesDocument Dim valCheck As NotesName Dim currentUser As New NotesName(session.UserName) Set db = session.CurrentDatabase Call db.sign(DBSIGN_DOC_ALL) Set view=db.Getview("config") Set viewNote=db.Getdocumentbyunid(view.Universalid) Set valCheck=New NotesName(viewNote.Getfirstitem("\$UpdatedBy").Values(Ubound(viewNote.Getfirstitem("\$UpdatedBy").Values)-1)) If valCheck.Organization<>currentUser.Organization Then Exit Sub Msgbox "The design of the Domino License Analysis Utility has been re-signed."+Chr(10)+Chr(10)+"the utility is now ready to use." Call ws.CurrentDocument.Close End Sub	

Frameset Information

Name:	fsInner
Last Modification:	01/11/2023 02:55:15 PM
Comment:	[Not Assigned]
Available to public access users:	No
Window Title Formula:	@DbTitle

HTML Code:

```
<frameset border="0" rows="*" cols="140,80%">
<frame name="Sidebar" frameborder="0" scrolling="no" noresize>
<frame name="Main" scrolling="auto">
</frameset>
```

Name:	fsLaunch
Last Modification:	01/11/2023 02:55:17 PM
Comment:	[Not Assigned]
Available to public access users:	No
Window Title Formula:	[Not Assigned]

HTML Code:

```
<frameset frameborder="0" border="0" rows="*" cols="*">
<frame frameborder="0">
</frameset>
```

Name:	fsReport
Last Modification:	01/11/2023 02:55:18 PM
Comment:	[Not Assigned]
Available to public access users:	No
Window Title Formula:	[Not Assigned]

HTML Code:

```
<frameset rows="" cols="">
<frame>
</frameset>
```

Name:

Last Modification:
Comment:
Available to public access users:
Window Title Formula:

HTML Code:

```
<frameset rows="" cols="">
<frame>
</frameset>
```

fsResults

01/11/2023 02:55:14 PM
[Not Assigned]
No
[Not Assigned]

Name:

Last Modification:
Comment:
Available to public access users:
Window Title Formula:

HTML Code:

```
<frameset frameborder="0" border="0" rows="" cols="">
<frame name="Sidebar" frameborder="0" scrolling="no" noresize>
</frameset>
```

fsSidebar

01/11/2023 02:55:15 PM
[Not Assigned]
No
[Not Assigned]

Name:

Last Modification:
Comment:
Available to public access users:
Window Title Formula:

HTML Code:

```
<frameset frameborder="0" border="0" rows="" cols="">
<frame name="Sidebar" frameborder="0" scrolling="no" noresize>
</frameset>
```

fsSidebar2

01/11/2023 02:55:15 PM
[Not Assigned]
No
[Not Assigned]

Name:

Last Modification:
Comment:
Available to public access users:
Window Title Formula:

HTML Code:

```
<frameset frameborder="0" border="0" rows="" cols="">
<frame name="Sidebar" frameborder="0" scrolling="no" noresize>
</frameset>
```

fsSidebar3

01/11/2023 02:55:15 PM
[Not Assigned]
No
[Not Assigned]

Name:

Last Modification:
Comment:
Available to public access users:

fsTools

01/11/2023 02:55:16 PM
[Not Assigned]
No

Window Title Formula:

[Not Assigned]

HTML Code:

```
<frameset rows="*" cols="*">
<frame>
</frameset>
```

Name:

Last Modification:

Comment:

Available to public access users:

Window Title Formula:

Launch

01/11/2023 02:55:12 PM

[Not Assigned]

No

@DbTitle

HTML Code:

```
<frameset frameborder="0" border="0" rows="100,*,45" cols="*">
<frame name="Header" frameborder="0" scrolling="no" noresize>
<frame name="Inner" frameborder="0" scrolling="auto" noresize>
<frame name="Footer" frameborder="0" scrolling="no" noresize>
</frameset>
```

Image Information

Name:

Last Modification:

Comment:

Images Across:

Images Down:

Colorize Grays:

actionbarblack.png

01/11/2023 02:55:18 PM

[Not Assigned]

1

1

No

Name:

Last Modification:

Comment:

Images Across:

Images Down:

Colorize Grays:

actionbarwhite.png

01/11/2023 02:55:12 PM

[Not Assigned]

1

1

No

Name:

Last Modification:

Comment:

Images Across:

Images Down:

Colorize Grays:

blue-bar.png

01/11/2023 02:55:17 PM

[Not Assigned]

1

1

No

Name:

Last Modification:

Comment:

Images Across:

Images Down:

Colorize Grays:

buttonbg.png

01/11/2023 02:55:16 PM

[Not Assigned]

1

1

No

Name:

Last Modification:
Comment:
Images Across:
Images Down:
Colorize Grays:

circle-dot.png

01/11/2023 02:55:17 PM
[Not Assigned]
1
1
No

Name:

Last Modification:
Comment:
Images Across:
Images Down:
Colorize Grays:

close.png

01/11/2023 02:55:12 PM
[Not Assigned]
1
1
No

Name:

Last Modification:
Comment:
Images Across:
Images Down:
Colorize Grays:

denyaccess.png

01/11/2023 02:55:14 PM
[Not Assigned]
1
1
No

Name:

Last Modification:
Comment:
Images Across:
Images Down:
Colorize Grays:

edit.png

01/11/2023 02:55:12 PM
[Not Assigned]
1
1
No

Name:

Last Modification:
Comment:
Images Across:
Images Down:
Colorize Grays:

export.png

01/11/2023 02:55:17 PM
[Not Assigned]
1
1
No

Name:

Last Modification:
Comment:
Images Across:
Images Down:
Colorize Grays:

history.png

01/11/2023 02:55:17 PM
[Not Assigned]
1
1
No

Name:

Last Modification:
Comment:
Images Across:
Images Down:

home.png

01/11/2023 02:55:12 PM
[Not Assigned]
1
1

Colorize Grays:

No

Name:

Last Modification:

Comment:

Images Across:

Images Down:

Colorize Grays:

information--square.png

01/11/2023 02:55:12 PM

[Not Assigned]

1

1

No

Name:

Last Modification:

Comment:

Images Across:

Images Down:

Colorize Grays:

info.png

01/11/2023 02:55:13 PM

[Not Assigned]

1

1

No

Name:

Last Modification:

Comment:

Images Across:

Images Down:

Colorize Grays:

loading.gif

01/11/2023 02:55:14 PM

[Not Assigned]

1

1

No

Name:

Last Modification:

Comment:

Images Across:

Images Down:

Colorize Grays:

logo.png

01/11/2023 02:55:13 PM

[Not Assigned]

1

1

No

Name:

Last Modification:

Comment:

Images Across:

Images Down:

Colorize Grays:

logo_header.png

01/11/2023 02:55:17 PM

[Not Assigned]

1

1

No

Name:

Last Modification:

Comment:

Images Across:

Images Down:

Colorize Grays:

logo_medium.png

01/11/2023 02:55:17 PM

[Not Assigned]

1

1

No

Name:

Last Modification:

Comment:

logo_small.png

01/11/2023 02:55:13 PM

[Not Assigned]

Images Across:	1
Images Down:	1
Colorize Grays:	No

Name:	logo_tiny.png
Last Modification:	01/11/2023 02:55:17 PM
Comment:	[Not Assigned]
Images Across:	1
Images Down:	1
Colorize Grays:	No

Name:	mobile--check.png
Last Modification:	01/11/2023 02:55:12 PM
Comment:	[Not Assigned]
Images Across:	1
Images Down:	1
Colorize Grays:	No

Name:	new.png
Last Modification:	01/11/2023 02:55:13 PM
Comment:	[Not Assigned]
Images Across:	1
Images Down:	1
Colorize Grays:	No

Name:	observation.png
Last Modification:	01/11/2023 02:55:18 PM
Comment:	[Not Assigned]
Images Across:	1
Images Down:	1
Colorize Grays:	No

Name:	print.png
Last Modification:	01/11/2023 02:55:12 PM
Comment:	[Not Assigned]
Images Across:	1
Images Down:	1
Colorize Grays:	No

Name:	refresh.png
Last Modification:	01/11/2023 02:55:14 PM
Comment:	[Not Assigned]
Images Across:	1
Images Down:	1
Colorize Grays:	No

Name:	save.png
--------------	-----------------

Last Modification:	01/11/2023 02:55:12 PM
Comment:	[Not Assigned]
Images Across:	1
Images Down:	1
Colorize Grays:	No

Name:	select.png
Last Modification:	01/11/2023 02:55:15 PM
Comment:	[Not Assigned]
Images Across:	1
Images Down:	1
Colorize Grays:	No

Name:	tasks.png
Last Modification:	01/11/2023 02:55:13 PM
Comment:	[Not Assigned]
Images Across:	1
Images Down:	1
Colorize Grays:	No

Name:	tools.png
Last Modification:	01/11/2023 02:55:16 PM
Comment:	[Not Assigned]
Images Across:	1
Images Down:	1
Colorize Grays:	No

Name:	tool_backup.png
Last Modification:	01/11/2023 02:55:16 PM
Comment:	[Not Assigned]
Images Across:	1
Images Down:	1
Colorize Grays:	No

Name:	tool_cleanup.png
Last Modification:	01/11/2023 02:55:16 PM
Comment:	[Not Assigned]
Images Across:	1
Images Down:	1
Colorize Grays:	No

Name:	tool_security_check.png
Last Modification:	01/11/2023 02:55:16 PM
Comment:	[Not Assigned]
Images Across:	1
Images Down:	1
Colorize Grays:	No

Name:	tool_settings_review.png
Last Modification:	01/11/2023 02:55:16 PM
Comment:	[Not Assigned]
Images Across:	1
Images Down:	1
Colorize Grays:	No

Name:	tool_tracking.png
Last Modification:	01/11/2023 02:55:16 PM
Comment:	[Not Assigned]
Images Across:	1
Images Down:	1
Colorize Grays:	No

Name:	tool_user_add.png
Last Modification:	01/11/2023 02:55:16 PM
Comment:	[Not Assigned]
Images Across:	1
Images Down:	1
Colorize Grays:	No

Name:	tool_user_remove.png
Last Modification:	01/11/2023 02:55:16 PM
Comment:	[Not Assigned]
Images Across:	1
Images Down:	1
Colorize Grays:	No

Name:	tool_web_check.png
Last Modification:	01/11/2023 02:55:16 PM
Comment:	[Not Assigned]
Images Across:	1
Images Down:	1
Colorize Grays:	No

Name:	twisty.gif
Last Modification:	01/11/2023 02:55:15 PM
Comment:	[Not Assigned]
Images Across:	2
Images Down:	1
Colorize Grays:	No

Name:	user_remove.png
Last Modification:	01/11/2023 02:55:13 PM
Comment:	[Not Assigned]
Images Across:	1
Images Down:	1

Colorize Grays:

No

Name:

Last Modification:

Comment:

Images Across:

Images Down:

Colorize Grays:

users.png

01/11/2023 02:55:13 PM

[Not Assigned]

1

1

No

Name:

Last Modification:

Comment:

Images Across:

Images Down:

Colorize Grays:

viewtwistie

01/11/2023 02:55:15 PM

[Not Assigned]

2

1

No

Name:

Last Modification:

Comment:

Images Across:

Images Down:

Colorize Grays:

view.png

01/11/2023 02:55:13 PM

[Not Assigned]

1

1

No

Name:

Last Modification:

Comment:

Images Across:

Images Down:

Colorize Grays:

warning.png

01/11/2023 02:55:17 PM

[Not Assigned]

1

1

No

Name:

Last Modification:

Comment:

Images Across:

Images Down:

Colorize Grays:

wizard.png

01/11/2023 02:55:17 PM

[Not Assigned]

1

1

No

Name:

Last Modification:

Comment:

Images Across:

Images Down:

Colorize Grays:

\$DBIcon

01/11/2023 02:55:12 PM

[Not Assigned]

1

1

No

Other Information

Type:

"About Application"

Last Modification:
JavaScript & HTML Code:
[None]
LotusScript Code:
[None]

Document
01/11/2023 02:55:13 PM