

## **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:

### **dlgCCXQuestion**

07/04/2023 10:52:19 AM  
[Not Assigned]  
Document  
No  
No  
No  
No  
No  
No  
No  
No  
No  
No  
No  
No  
No  
[None Assigned]  
All Users  
All Users

### **Subcomponents:**

[None]

### **JavaScript & HTML Code:**

[None]

### **LotusScript Code:**

[None]

### **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**

07/04/2023 10:52:17 AM  
[Not Assigned]  
Document  
No  
No  
No  
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:

#### **DAList**

Text  
Comma, Semicolon, New Line  
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:	<pre>val := @DbLookup("" : "NoCache"; "" ; "config"; "config"; "DAList");ret := @if(@IsError(val); "&lt; Error Retrieving Server Names &gt;"; @Word(val; "~"; 1)); @Trim(@If(DAListExternal = ""; ret; @Replace(ret; DAListExternal; "")));</pre>
Is Scripted:	No
<b>Field:</b>	<b>Identifier</b>
Datatype:	Keywords
Input Multi-Value Separator(s):	Comma
Display Multi-Value Separator:	Comma
Help Description:	[Not Assigned]
Field Type:	Editable
Keyword User Interface:	Check Boxes
Allowable Keywords:	Additional Directory d Org/OrgUnit in Notes Name o
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Is Scripted:	No
<b>Field:</b>	<b>DAListExternal</b>
Datatype:	Keywords
Help Description:	[Not Assigned]
Field Type:	Editable
Keyword User Interface:	Standard
Allow Values Not In List:	No
Allowable Keywords Formula:	<pre>values := DAList;list := @if(@IsError(values); ""; @ReplaceSubstring(@Replace Substring(values; " "; "!"); "~"; " "));list;</pre>
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Is Scripted:	No
Hide When:	@IsNotMember("d"; Identifier);
<b>Field:</b>	<b>OrgsExternal</b>
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:	<pre>val := @DbLookup("" : "NoCache"; "" ; "config"; "config"; "Orgs");@If(@IsError(val); "&lt; Error Retrieving Orgs &gt;"; val);</pre>

Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Is Scripted:	No
Hide When:	@IsNotMember("o"; Identifier);
<b>JavaScript &amp; HTML Code:</b>	
[None]	
<b>LotusScript Code:</b>	
[None]	
<b>Name:</b>	<b>dlgUserInfo</b>
Last Modification:	07/04/2023 10:52:17 AM
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
<b>Subcomponents:</b>	
<b>Field:</b>	<b>CompanyName</b>
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
<b>Field:</b>	<b>HCLAccount</b>
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
<b>Field:</b>	<b>Name</b>
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
<b>Field:</b>	<b>Email</b>
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
<b>Field:</b>	<b>Phone</b>
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
<b>JavaScript &amp; HTML Code:</b>	
[None]	
<b>LotusScript Code:</b>	
[None]	
<b>Name:</b>	<b>fmGroup</b>
Alias:	Group
Last Modification:	07/04/2023 10:52:17 AM
Comment:	[Not Assigned]
Type:	Document
Window Title Formula:	"Deny Access Group 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
<b>Subcomponents:</b>	
<b>Field:</b>	<b>NABNOTEUPDATED</b>
Datatype:	Time
Time Format:	07/04/2023 10:53 AM
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
<b>Action:</b>	<b>Spacer</b>
<b>Action:</b>	<b>FormName</b>
<b>Action:</b>	<b>Spacer</b>
<b>Action:</b>	<b>Close</b>

Formula:	@Command([FileCloseWindow]);
Action:	Print
Formula:	@Command([FilePrint]);
<b>JavaScript &amp; HTML Code:</b>	
[None]	
<b>LotusScript Code:</b>	
(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]	
<b>Name:</b>	<b>fmHistory</b>
Alias:	History
Last Modification:	07/04/2023 10:52:17 AM
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
<b>Subcomponents:</b>	
Field:	<b>SaveOptions</b>
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

<b>Field:</b>	<b>PersonDocCount</b>
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:	@ThisValue
Is Scripted:	No
<b>Field:</b>	<b>GroupDocCount</b>
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:	@ThisValue
Is Scripted:	No
<b>Field:</b>	<b>DAUserCount</b>
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:	@ThisValue
Is Scripted:	No
<b>Field:</b>	<b>LicenseTrackingDocCount</b>
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:	@ThisValue
Is Scripted:	No
<b>Field:</b>	<b>ValidUserCount</b>
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:	@ThisValue
Is Scripted:	No
<b>Field:</b>	<b>ValidUserCountCCB</b>
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:	@ThisValue
Is Scripted:	No
<b>Field:</b>	<b>ValidUserCountCCXActive</b>
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:	@ThisValue
Is Scripted:	No
<b>Field:</b>	<b>ValidUserCountCCXInactive</b>
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:	@ThisValue
Is Scripted:	No
<b>Field:</b>	<b>MailUserCount</b>
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:	@ThisValue
Is Scripted:	No
<b>Field:</b>	<b>EntitlementTrackingServers</b>
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
<b>Field:</b>	<b>UserReportLocation</b>
Datatype:	Text
Help Description:	[Not Assigned]
Field Type:	Computed
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Formula:	@ThisValue
Is Scripted:	No
<b>Field:</b>	<b>DAScanError</b>
Datatype:	Text
Help Description:	[Not Assigned]
Field Type:	Computed
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Formula:	@ThisValue
Is Scripted:	No
<b>Field:</b>	<b>LicenseTrackingServersMissing</b>
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
<b>Field:</b>	<b>LDAPDA</b>
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:**

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

**Hotspot**

Type:  
Caption:  
Formula:

**Hotspot**

Type:  
Caption:  
Formula:

**Hotspot**

Type:  
Caption:  
Formula:

**Hotspot**

Type:  
Caption:  
Formula:

**GetInfoCancelled**

Text  
[Not Assigned]  
Editable  
No  
Disabled  
No  
No

Action Hotspot  
CCB

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

Action Hotspot  
Users

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

Action Hotspot  
CCX

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

Action Hotspot  
Users

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

**Hotspot**

Type:  
Caption:  
Formula:

Action Hotspot  
Inactive CCX

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

**Hotspot**

Type:  
Caption:  
Formula:

Action Hotspot  
Users

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

**Hotspot**

Type:  
Caption:  
Formula:

Action Hotspot  
Person

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

**Hotspot**

Type:  
Caption:  
Formula:

Action Hotspot  
Documents

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

**Hotspot**

Type:  
Caption:  
Formula:

Action Hotspot  
Mail

```
@Do(@SetTargetFrame("Sideb  
ar");  
@Command([OpenFrameset];  
"fsSidebar3"));
```

**Hotspot**

Type:  
Caption:  
Formula:

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

Action Hotspot  
Users

**Hotspot**

Type:  
Caption:  
Formula:

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

Action Hotspot  
Deny Access

**Hotspot**

Type:  
Caption:  
Formula:

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

Action Hotspot  
Groups

**Hotspot**

Type:  
Caption:  
Formula:

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

Action Hotspot  
Deny Access

**Hotspot**

Type:  
Caption:  
Formula:

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

Action Hotspot  
Users

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

**Field:**

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

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

**AdminServer**

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

No

**ServerList**

Keywords  
 [Not Assigned]  
 Editable  
 Standard  
 No  
 val := ServerList;  
 @If(@IsError(val); "< Error  
 Retrieving Server Names >";  
 @Word(val; "|"; 1) + " (" +  
 @Word(val; "|"; 2) + ")");

No

Disabled

No

""

No

**EntitlementTrackingServersDis  
p**

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

**DAInfo**

Text  
 [Not Assigned]  
 Computed when composed  
 No

[illegible]

onClick  
**Field AdminServer**  
onClick  
**Field ServerList**  
onClick  
onClick  
onClick  
**Field DAList**  
onClick  
[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:  
Activate Objects When Editing:  
Activate Objects When Reading:  
Document Encryption Keys:  
Composed Documents May Be Read By:  
Form May Be Composed By:

## Subcomponents:

Field:

## fmLaunch

07/04/2023 10:52:17 AM  
[Not Assigned]  
Document  
@DbTitle  
No  
No  
No  
No  
No  
No  
No  
No  
No  
No  
No  
No  
No  
No  
[None Assigned]  
All Users  
All Users

\$Buttons

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:	"1"
Is Scripted:	No
<b>Field:</b>	<b>\$Steps</b>
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:	"1"
Is Scripted:	No
<b>Field:</b>	<b>SaveOptions</b>
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
<b>Field:</b>	<b>ConfigCheck</b>
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"; @DbName; "config"; "config"; "ScansComplete"); @If(@IsError(val); "" ; val = "" ; "" ; val);
Is Scripted:	No
<b>Field:</b>	<b>PersonDocCount</b>
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:**

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:  
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:

**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  
Disabled  
No  
val := @DbLookup("" :  
"NoCache"; "" ; "config"; "config";  
@ThisName);@If(@IsError(val);  
-20; @IsNumber(val); val; -1);  
No

**LicenseTrackingDocCount**

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

**ValidUserCount**

Number  
General  
No  
No  
No  
[Not Assigned]  
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
<b>Field:</b>	<b>EntitlementTrackingServers</b>
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
<b>Field:</b>	<b>UserReportLocation</b>
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
<b>Field:</b>	<b>DAScanError</b>
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
<b>Field:</b>	<b>LicenseTrackingServersMissin g</b>
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
<b>Hotspot</b>	Button
Type:	PerformScan
Caption:	scanType := @If(Step = 2; "Config"; Step = 3; "User"; Step = 4; "DAGroups"; "Entitlement"); FIELD Scan := scanType;
Formula:	@Command([RefreshHideForm ulas]); @PostedCommand([ToolsRun Macro]; "PullInfo" + scanType);
<b>Hotspot</b>	Button
Type:	MakeHistory
Caption:	@Command([ToolsRunMacro]; "MakeHistory");
Formula:	
<b>Hotspot</b>	Button
Type:	FirstNext
Caption:	Yes, see below
Is Scripted:	<b>Step</b>
<b>Field:</b>	Number
Datatype:	General
Number Format:	No
Percentage (value * 100)%:	No
Parentheses on Negative Numbers:	No
Punctuated at Thousands:	[Not Assigned]
Help Description:	Editable
Field Type:	No
Sign When Mailing/In Section:	Disabled
Encryption:	No
Update Requires Editor Access:	No
Default Value Formula:	1
Is Scripted:	No
<b>Field:</b>	<b>Scan</b>
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
<b>Field:</b>	<b>UserScan</b>
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
<b>Field:</b>	<b>DAScan</b>
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
<b>Field:</b>	<b>EntitlementScan</b>
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
<b>Field:</b>	<b>ver</b>
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:	@TextToNumber(@Version);
Is Scripted:	No
<b>Hotspot</b>	
Type:	Link
Caption:	HCL Domino CCB License Agreement
Associated Data:	<a href="https://www.hcltechsw.com/wps/wcm/connect/c6f1e7e3-d6d3-436d-a95a-44c5347dca1a/HCL+Domino+Complete+Collaboration+12.0.1.pdf?MOD=AJPERES&amp;CONVERT_TO=url&amp;CACHEID=ROOTWORKSPACE-c6f1e7e3-d6d3-436d-a95a-44c5347dca1a-nSOi.Tg">https://www.hcltechsw.com/wps/wcm/connect/c6f1e7e3-d6d3-436d-a95a-44c5347dca1a/HCL+Domino+Complete+Collaboration+12.0.1.pdf?MOD=AJPERES&amp;CONVERT_TO=url&amp;CACHEID=ROOTWORKSPACE-c6f1e7e3-d6d3-436d-a95a-44c5347dca1a-nSOi.Tg</a>
<b>Hotspot</b>	
Type:	Link
Caption:	User Report Tool
Associated Data:	

**Hotspot**

Type:

Caption:

Associated Data:

[https://support.hcltechsw.com/csm?id=kb\\_article&sysparm\\_article=KB0095328](https://support.hcltechsw.com/csm?id=kb_article&sysparm_article=KB0095328)

Link

HCL Domino Administration  
Documentation

<https://help.hcltechsw.com/domino/12.0.0/admin/index.html>

**Hotspot**

Type:

Caption:

Associated Data:

Link

Entitlement Tracking System

[https://help.hcltechsw.com/domino/12.0.0/admin/admn\\_entitlementtracking.html](https://help.hcltechsw.com/domino/12.0.0/admin/admn_entitlementtracking.html)

**Hotspot**

Type:

Caption:

**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:

Action Hotspot

Start Environment Scan Now

**AdminServer**

Text

[Not Assigned]

Computed when composed

No

Disabled

No

@ThisValue

No

**ServerList**

Keywords

[Not Assigned]

Editable

Standard

No

```
val := @DbLookup("" :  
"NoCache"; "" ; "config"; "config";  
"ServerList");@If(@IsError(val);  
"< Error Retrieving Server  
Names >"; @Word(val; "|"; 1) + "  
(" + @Word(val; "|"; 2) + ")");
```

No

Disabled

No

""

No

**DAInfo**

Text

[Not Assigned]

Computed

No

Disabled

No

```
val1 := @DbLookup("" :  
"NoCache"; "" ; "config"; "config";  
"DAName");val2 :=
```

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

Is Scripted:  
Hide When:

**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:

Is Scripted:  
Hide When:

**Hotspot**  
Type:  
Caption:

**Hotspot**  
Type:  
Caption:  
Formula:

```
@DbLookup( "" : "NoCache"; "";  
"config"; "config";  
"DAFilename");  
@If(@IsError(val1); "< Error  
Retrieving Directory Assistance  
Information >"; val1 = "";  
"Directory Assistance is not  
configured."; val1 + " (" + val2 +  
")");  
No
```

**DAListExternal**

Text  
[Not Assigned]  
Computed  
No  
Disabled  
No  
val := @DbLookup( "" :  
"NoCache"; "" ; "config"; "config";  
@ThisName);@If(@IsError(val);  
"< Error Retrieving Server  
Names >"; @Word(val; "~"; 1));  
No

DAListExternal = "" |  
AdminServer = "";

**DAListDisplay**

Text  
Comma, Semicolon, New Line  
New Line  
[Not Assigned]  
Computed  
No  
Disabled  
No  
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

DAInfo = "" | AdminServer = "";

Action Hotspot  
Start Users Scan Now

Action Hotspot  
Person Documents

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

**Hotspot**

Type:  
Caption:

**Hotspot**

Type:  
Caption:  
Formula:

**Hotspot**

Type:  
Caption:  
Formula:

**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:

**Hotspot**

Type:  
Caption:

**Hotspot**

Type:  
Caption:

```
"fsSidebar2"));
@Do(@SetTargetFrame("Main"
);
@PostedCommand([OpenView]
; "UserInfo"));
```

Action Hotspot  
Start Deny Access Scan Now

Action Hotspot  
Deny Access Groups

```
@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

**EntitlementTrackingServersDis  
p**

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

Action Hotspot  
Start User Report Tool Scan  
Now

Action Hotspot  
Show Previous Results

Formula:	@SetTargetFrame("Inner"); @Command([OpenFrameset]; "fsInner");
<b>Hotspot</b> Type: Caption: Is Scripted:	Action Hotspot Next Yes, see below
<b>Hotspot</b> Type: Caption: Formula:	Action Hotspot Cancel @Command([CloseWindow]);
<b>Hotspot</b> Type: Caption: Is Scripted:	Action Hotspot Back Yes, see below
<b>Hotspot</b> Type: Caption: Is Scripted:	Action Hotspot Next Yes, see below
<b>Hotspot</b> Type: Caption: Formula:	Action Hotspot Cancel @Command([CloseWindow]);
<b>Hotspot</b> Type: Caption: Is Scripted:	Action Hotspot Back Yes, see below
<b>Hotspot</b> Type: Caption: Is Scripted:	Action Hotspot Next Yes, see below
<b>Hotspot</b> Type: Caption: Formula:	Action Hotspot Cancel @Command([CloseWindow]);
<b>Hotspot</b> Type: Caption: Is Scripted:	Action Hotspot Back Yes, see below
<b>Hotspot</b> Type: Caption: Is Scripted:	Action Hotspot Next Yes, see below
<b>Hotspot</b> Type: Caption: Formula:	Action Hotspot Cancel @Command([CloseWindow]);
<b>Hotspot</b> Type: Caption: Is Scripted:	Action Hotspot Back Yes, see below
<b>Hotspot</b> Type: Caption: Is Scripted:	Action Hotspot Next Yes, see below
<b>Hotspot</b> Type: Caption: Formula:	Action Hotspot Cancel @Command([CloseWindow]);
<b>Hotspot</b> Type: Caption: Is Scripted:	Action Hotspot Back Yes, see below
<b>Hotspot</b> Type: Caption: Is Scripted:	Action Hotspot Next Yes, see below

Type:	Action Hotspot
Caption:	Cancel
Formula:	@Command([CloseWindow]);
<b>Hotspot</b>	
Type:	Action Hotspot
Caption:	Back
Is Scripted:	Yes, see below
<b>Hotspot</b>	
Type:	Action Hotspot
Caption:	Next
Is Scripted:	Yes, see below
<b>Hotspot</b>	
Type:	Action Hotspot
Caption:	Cancel
Formula:	@Command([CloseWindow]);
<b>Hotspot</b>	
Type:	Action Hotspot
Caption:	View Results
Formula:	@SetTargetFrame("Inner"); @Command([OpenFrameset]; "fsInner");
<b>Subform:</b>	<b>sfmlInfoOverlays</b>
<b>JavaScript &amp; HTML Code:</b>	
<b>Action Hotspot</b>	
Client onClick	
document.forms[0].Timer.click();	
setTimeout(document.forms[0].PerformScan.click(),1);	
onClick	
<b>Field AdminServer</b>	
onClick	
<b>Field ServerList</b>	
onClick	
onClick	
<b>Action Hotspot</b>	
Client onClick	
document.forms[0].Timer.click();	
setTimeout(document.forms[0].PerformScan.click(),1);	
onClick	
<b>Action Hotspot</b>	
Client onClick	
document.forms[0].Timer.click();	
setTimeout(document.forms[0].PerformScan.click(),1);	
onClick	
onClick	
<b>Action Hotspot</b>	
Client onClick	
document.forms[0].Timer.click();	
setTimeout(document.forms[0].PerformScan.click(),1);	
onClick	
onClick	
<b>Action Hotspot</b>	
Client onClick	
document.forms[0].Timer.click();	
setTimeout(document.forms[0].PerformScan.click(),1);	
<b>LotusScript Code:</b>	
(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
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
        Set source=Nothing
        Exit Sub
    End If
    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=""
        End If
    End If
End Sub

```

```

        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
    If openAdmin.Getfirstitem(fldName).text<>"" Then
        openGetAdminValue=openAdmin.Getfirstitem(fldName).Values
    End If
End If
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=openadb.Getview("config")
    Set openadmin=openview.Getdocumentbykey("config",t)
    If openadmin Is Nothing Then
        Set openadmin=openadb.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()
On Error Resume Next
Dim encStr As String
Dim adminVal As Variant
Dim nl As String
Dim wd As Integer
Dim x As Integer
nl=Chr(10)
wd=GetNum
' Date when date was encoded
encStr="|Date:"|+Cstr(Now)+"|",|+nl
' Authorized User Count

```

```

adminVal=openGetAdminValue("ValidUserCount")
If Not Isarray(adminVal) Then
    encStr=encStr+"AU":"0",|+nl
Elseif Cstr(adminVal(0))="" Then
    encStr=encStr+"AU":"0",|+nl
Else
    encStr=encStr+"AU:"|+Cstr(adminVal(0))+|",|+nl
End If
' CCB User count
adminVal=openGetAdminValue("ValidUserCountCCB")
If Not Isarray(adminVal) Then
    encStr=encStr+"CCB":"0",|+nl
Elseif Cstr(adminVal(0))="" Then
    encStr=encStr+"CCB":"0",|+nl
Else
    encStr=encStr+"CCB:"|+Cstr(adminVal(0))+|",|+nl
End If
' CCX Active User count
adminVal=openGetAdminValue("ValidUserCountCCXActive")
If Not Isarray(adminVal) Then
    encStr=encStr+"CCXA":"0",|+nl
Elseif Cstr(adminVal(0))="" Then
    encStr=encStr+"CCXA":"0",|+nl
Else
    encStr=encStr+"CCXA:"|+Cstr(adminVal(0))+|",|+nl
End If
' CCX Inactive User count
adminVal=openGetAdminValue("ValidUserCountCCXInactive")
If Not Isarray(adminVal) Then
    encStr=encStr+"CCXI":"0",|+nl
Elseif Cstr(adminVal(0))="" Then
    encStr=encStr+"CCXI":"0",|+nl
Else
    encStr=encStr+"CCXI:"|+Cstr(adminVal(0))+|",|+nl
End If
' Known Guest User count
adminVal=openGetAdminValue("ValidUserCountGuest")
If Not Isarray(adminVal) Then
    encStr=encStr+"KG":"0",|+nl
Elseif Cstr(adminVal(0))="" Then
    encStr=encStr+"KG":"0",|+nl
Else
    encStr=encStr+"KG:"|+Cstr(adminVal(0))+|",|+nl
End If
' Total Person Document count
adminVal=openGetAdminValue("PersonDocCount")
If Not Isarray(adminVal) Then
    encStr=encStr+"PD":"0",|+nl
Elseif Cstr(adminVal(0))="" Then
    encStr=encStr+"PD":"0",|+nl
Else
    encStr=encStr+"PD:"|+Cstr(adminVal(0))+|",|+nl
End If
' Notes Mail User count
adminVal=openGetAdminValue("MailUserCount")
If Not Isarray(adminVal) Then

```

```

        encStr=encStr+"MU":"0",|+nl
    ElseIf Cstr(adminVal(0))="" Then
        encStr=encStr+"MU":"0",|+nl
    Else
        encStr=encStr+"MU":|+Cstr(adminVal(0))+|",|+nl
    End If
    ' Deny Access Group count
    adminVal=openGetAdminValue("GroupDocCount")
    If Not Isarray(adminVal) Then
        encStr=encStr+"DAG":"0",|+nl
    ElseIf Cstr(adminVal(0))="" Then
        encStr=encStr+"DAG":"0",|+nl
    Else
        encStr=encStr+"DAG":|+Cstr(adminVal(0))+|",|+nl
    End If
    ' Deny Access User count
    adminVal=openGetAdminValue("DAUserCount")
    If Not Isarray(adminVal) Then
        encStr=encStr+"DAU":"0",|+nl
    ElseIf Cstr(adminVal(0))="" Then
        encStr=encStr+"DAU":"0",|+nl
    Else
        encStr=encStr+"DAU":|+Cstr(adminVal(0))+|",|+nl
    End If
    ' License tracking document count
    adminVal=openGetAdminValue("LicenseTrackingDocCount")
    If Not Isarray(adminVal) Then
        encStr=encStr+"LTD":"0"|
    ElseIf Cstr(adminVal(0))="" Then
        encStr=encStr+"LTD":"0"|
    Else
        encStr=encStr+"LTD":|+Cstr(adminVal(0))+|"|
    End If
    ' Indication if LDAP directories are in use
    adminVal=openGetAdminValue("LDAPDA")
    If Isarray(adminVal) Then If Cstr(adminVal(0))<>"" Then encStr=encStr+"LDAP":"LDAP Directory In Use a
    For x=1 To wd
        encStr=Encode(encStr)
    Next
    openSetAdminValue "Signature",Cstr(wd)+encStr
    openSetAdminValue "SignatureDate",Now
    openSetAdminValue "SignatureUser",openSession.Username
End Sub
Function Encode (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)
    Encode = 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
Sub SetCounts()
    Dim nav As NotesViewNavigator
    Dim dc As NotesDocumentCollection
    Dim keys(1) As String
    Set openView=openDB.GetView("(users)")
    Call openView.Refresh()
    openView.Autoupdate=False
    Set nav=openView.CreateViewNav
    openSetAdminValue "PersonDocCount",nav.Count
    Set openView=openDB.GetView("(dagroups)")
    Call openView.Refresh()
    openView.Autoupdate=False
    Set nav=openView.CreateViewNav
    openSetAdminValue "GroupDocCount",nav.Count
    Set openView=openDB.GetView("(dauser)")
    Call openView.Refresh()
    openView.Autoupdate=False
    Set nav=openView.CreateViewNav
    openSetAdminValue "DAUserCount",nav.Count
    Set openView=openDB.GetView("(mailuser)")
    Call openView.Refresh()
    openView.Autoupdate=False
    Set nav=openView.CreateViewNav
    openSetAdminValue "MailUserCount",nav.Count
    Set openView=openDB.GetView("(valid)")
    Call openView.Refresh()
    openView.Autoupdate=False
    Set nav=openView.CreateViewNav
    openSetAdminValue "ValidUserCount",nav.Count
    keys(0)="all"
    keys(1)="ccb"
    Set openView=openDB.GetView("(counts)")
    Call openView.Refresh()
    openView.Autoupdate=False
    Set dc=openView.Getalldocumentsbykey(keys,True)
    openSetAdminValue "ValidUserCountCCB",dc.Count
    keys(1)="ccx active"
    Set dc=openView.Getalldocumentsbykey(keys,True)
    openSetAdminValue "ValidUserCountCCXActive",dc.Count
    keys(1)="ccx inactive"
    Set dc=openView.Getalldocumentsbykey(keys,True)
    openSetAdminValue "ValidUserCountCCXInactive",dc.Count
    keys(1)="known guest"
    Set dc=openView.Getalldocumentsbykey(keys,True)
    openSetAdminValue "ValidUserCountGuest",dc.Count
End Sub
Function GetNum() As Integer
    Dim randomNumber As Integer
    randomNumber = Int((3 * Rnd) + 2)
    GetNum = randomNumber

```

```

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)
    SetCounts
    If source.Document.Step(0)=7 Then
        openSetAdminValue "AllScansComplete","true"
        CreateDigitalSignature
    End If
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) Next
Sub Click(Source As Button)
    Dim agent As NotesAgent
    Set agent=openDB.GetAgent("MakeHistory")
    Call agent.Run
    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)
End Sub

```

```

        Call openui.Refresh
    End Sub
    (Hotspot) Next
    Sub Click(Source As Button)
        On Error Goto errHandle
        If openGetAdminValue("AskExternal")(0)<>"" Then
            Dim dialog As New NotesDocument(opendb)
            If openWS Is Nothing Then Set openWS=NewNotesUIWorkspace
            If openWS.Dialogbox("dlgCCXQuestion", t, t, f, f, f, f, "", dialog, t, f, t) Then
                openDialog:
                    If openWS.Dialogbox("dlgExternalQA", t, t, f, f, f, f, "", dialog, t, f, t) Then
                        If dialog.DAListExternal(0)="" And dialog.OrgsExternal(0)="" Then
                            If MsgBox("You did not select any values."+Chr(10)+Chr(10)+"Would you like to select values?", vbYesNo) = vbYes Then
                                openSetAdminValue "DAListExternal", dialog.DAListExternal(0)
                                openSetAdminValue "OrgsExternal", dialog.Getfirstitem("OrgsExternal").Values
                            End If
                        End If
                    End If
                End If
            End If
            stepVal="3"
            opennote.~$Buttons =stepVal
            opennote.~$Steps =stepVal
            opennote.Step=Cint(stepVal)
            Call openui.Refresh
        Exit Sub
    errHandle:
        MsgBox "ERROR: "+Error+Chr(10)+Chr(10)+"#: "+Cstr(Err)+"", line: "+Cstr(Erl)
    Exit Sub
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)
    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)
    openSetAdminValue "AllScansComplete","true"
    Call openui.Refresh
End Sub

```

**Name:**

Alias:

Last Modification:

Comment:

Type:

Window Title Formula:

Include in Compose Menu:

**fmLog**

config

07/04/2023 10:52:17 AM

[Not Assigned]

Document

"HCL Domino License Analysis -  
Results From " +

@Text(@Created; "D0T1Z0");

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:

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:

### JavaScript & HTML Code:

[None]

### LotusScript Code:

#### (Globals)

Option Public  
 Option Declare  
 [None]

#### 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

#### Print

@Command([FilePrint]);

**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:

**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:

**fmObservations**

Observations  
07/04/2023 10:52:17 AM  
[Not Assigned]  
Document  
"HCL Domino License Analysis -  
Results From " +  
@Text(@Created; "D0T1Z0");  
No  
No  
No  
No  
No  
No  
No  
No  
No  
No  
[None Assigned]  
All Users  
All Users

**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

Default Value Formula:	""
Is Scripted:	No
<b>Field:</b>	<b>Observation</b>
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 := ObservationsDisp;@If(val = "", "", @Word(val, "~", 2));
Is Scripted:	No
<b>Hotspot</b>	
Type:	Link
Caption:	Click here for full instructions on enabling License Tracking f
Associated Data:	<a href="https://help.hcltechsw.com/domin/12.0.0/admin/conf_licensetracking_t.html">https://help.hcltechsw.com/domin/12.0.0/admin/conf_licensetracking_t.html</a>
<b>Action:</b>	<b>Spacer</b>
<b>Action:</b>	<b>FormName</b>
<b>Action:</b>	<b>Spacer</b>
<b>Action:</b>	<b>Close</b>
Formula:	@Command([CloseWindow]);
<b>Action:</b>	<b>Print</b>
Formula:	@Command([FilePrint]);
<b>JavaScript &amp; HTML Code:</b>	
[None]	
<b>LotusScript Code:</b>	
(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]	
<b>Name:</b>	<b>fmResults</b>
Last Modification:	07/04/2023 10:52:19 AM
Comment:	[Not Assigned]

Type:	Document
Window Title Formula:	@DbTitle
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:

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

### SaveOptions

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

### 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:

### 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

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:

### GroupDocCount

Number  
 General  
 No  
 No  
 No  
 [Not Assigned]  
 Computed  
 No  
 Disabled  
 No  
 val := @DbColumn("" :  
 "nocache"; "" ; "dagroups"; 1);

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:  
Encryption:  
Update Requires Editor Access:  
Formula:

Is Scripted:  
**Field:**  
Datatype:  
Number Format:  
Percentage (value \* 100)%:  
Parentheses on Negative Numbers:

@If(@IsError(val); -1;  
@Elements(val));  
No  
**DAUserCount**  
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  
**LicenseTrackingDocCount**  
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  
**ValidUserCount**  
Number  
General  
No  
No  
No  
[Not Assigned]  
Computed  
No  
Disabled  
No  
val := @DbLookup("" :  
"NoCache"; "" ; "config"; "config";  
@ThisName);@If(@IsError(val);  
-10101; val);  
No  
**ValidUserCountCCB**  
Number  
General  
No  
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
<b>Field:</b>	<b>ValidUserCountCCXActive</b>
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"; "ValidUserCountCCXActive"); @If(@IsError(val); -10101; val);
Is Scripted:	No
<b>Field:</b>	<b>ValidUserCountCCXInactive</b>
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
<b>Field:</b>	<b>ValidUserCountGuest</b>
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";

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:  
Input Multi-Value Separator(s):  
Display Multi-Value Separator:  
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:

@ThisName);@If(@IsError(val);  
-10101; val);  
No  
**MailUserCount**  
Number  
General  
No  
No  
No  
[Not Assigned]  
Computed  
No  
Disabled  
No  
val := @DbLookup("" :  
"NoCache"; "" ; "config"; "config";  
@ThisName);@If(@IsError(val);  
-10101; val);  
No  
**EntitlementTrackingServers**  
Text  
Comma, Semicolon, New Line  
New Line  
[Not Assigned]  
Computed when composed  
No  
Disabled  
No  
val := @DbLookup("" :  
"NoCache"; "" ; "config"; "config";  
@ThisName);@If(@IsError(val);  
"< Error Retrieving Server  
Names >"; @Word(val; "|"; 1));  
No  
**UserReportLocation**  
Text  
[Not Assigned]  
Computed  
No  
Disabled  
No  
val := @DbLookup("" :  
"NoCache"; "" ; "config"; "config";  
@ThisName);@If(@IsError(val);  
"< Error Retrieving Server  
Names >"; @Word(val; "|"; 1));  
No  
**DAScanError**  
Text  
[Not Assigned]  
Computed  
No  
Disabled  
No  
check := @DbLookup("" :  
"NoCache"; "" ; "config"; "config";

Is Scripted:  
**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:

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:

#### **Hotspot**

Type:  
Caption:  
Is Scripted:

#### **Hotspot**

Type:  
Caption:  
Formula:

```
"DAScanError");  
@If(@IsError(check); ""; check  
= ""; "", "Note: there was an error  
scanning the additional Domino  
Directories");  
No
```

#### **LicenseTrackingServersMissin g**

Text  
Comma, Semicolon, New Line  
New Line  
[Not Assigned]  
Computed when composed  
No  
Disabled  
No  
val := @DbLookup("" :  
"NoCache"; ""; "config"; "config";  
@ThisName);@If(@IsError(val);  
"< 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

#### **GetInfoCancelled**

Text  
[Not Assigned]  
Editable  
No  
Disabled  
No  
No

Button  
GetInfo  
Yes, see below

Button  
OpenReport  
REM  
{@If(GetInfoCancelled="";@Suc  
cess;@Return(@False));  
@SetTargetFrame("Main");  
@Command([OpenFrameset];"f  
sReport"));@True;



**Hotspot**

Type:  
Caption:  
Formula:

Action Hotspot  
CCB

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

**Hotspot**

Type:  
Caption:  
Formula:

Action Hotspot  
Users

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

**Hotspot**

Type:  
Caption:  
Formula:

Action Hotspot  
CCX

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

**Hotspot**

Type:  
Caption:  
Formula:

Action Hotspot  
Users

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

**Hotspot**

Type:  
Caption:  
Formula:

Action Hotspot  
Inactive CCX

```
@Do(@SetTargetFrame("Sideb  
ar");  
@Command([OpenFrameset];  
"fsSidebar2"));
```

**Hotspot**

Type:  
Caption:  
Formula:

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

Action Hotspot  
Users

**Hotspot**

Type:  
Caption:  
Formula:

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

Action Hotspot  
Person

**Hotspot**

Type:  
Caption:  
Formula:

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

Action Hotspot  
Documents

**Hotspot**

Type:  
Caption:  
Formula:

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

Action Hotspot  
Mail

**Hotspot**

Type:  
Caption:  
Formula:

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

Action Hotspot  
Users

**Hotspot**

Type:  
Caption:  
Formula:

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

Action Hotspot  
Deny Access

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

Action Hotspot  
Groups

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

Action Hotspot  
Deny Access

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

**Hotspot**

Type:  
Caption:  
Formula:

**Hotspot**

Type:  
Caption:  
Formula:

**Hotspot**

Type:  
Caption:  
Formula:

Action Hotspot  
View Report

```
@SetEnvironment("PersonDoc  
Count";  
@Text(PersonDocCount));  
@SetEnvironment("DAUserCou  
nt"; @Text(DAUserCount));  
@SetEnvironment("ValidUserC  
ount"; @Text(ValidUserCount));  
@Command([ToolsRunMacro];  
"(ViewReportLS)");  
@PostedCommand([ToolsRun  
Macro]; "(ViewReportFM)");
```

**AdminServer**

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

**ServerList**

Keywords  
[Not Assigned]  
Editable  
Standard  
No  
val := @DbLookup("" :  
"NoCache"; "" ; "config"; "config";  
"ServerList");@If(@IsError(val);  
"< Error Retrieving Server  
Names >"; @Word(val; "|"; 1) + "  
(" + @Word(val; "|"; 2) + ")");  
No  
Disabled  
No  
""  
No

**EntitlementTrackingServersDis**

p  
Keywords  
[Not Assigned]  
Editable  
Standard  
No  
EntitlementTrackingServers  
No

**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:  
Keyword User Interface:  
Allow Values Not In List:  
Allowable Keywords Formula:  
Sign When Mailing/In Section:

Encryption:	Disabled
Update Requires Editor Access:	No
Is Scripted:	No
<b>Field:</b>	<b>DALInfo</b>
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
<b>Field:</b>	<b>DAListExternal</b>
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 Server Names >"; @Word(val; "~"; 1));
Is Scripted:	No
Hide When:	DAListExternal = "";
<b>Field:</b>	<b>DAList</b>
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"; "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:	DAInfo = "";
<b>Field:</b>	<b>ObservationsDisp</b>
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"; "Observations"); @If(@IsError(val); ""; @Word(val; " "; 1));
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Is Scripted:	No
<b>Subform:</b>	<b>sfmlInfoOverlays</b>
<b>JavaScript &amp; HTML Code:</b>	
onClick	
onClick	
onClick	
onClick	
onClick	
onClick	
onClick	
onClick	
onClick	
onClick	
<b>Field AdminServer</b>	
onClick	
<b>Field ServerList</b>	
onClick	
onClick	
onClick	
<b>Field DAList</b>	
onClick	
[None]	
<b>LotusScript Code:</b>	
<b>(Globals)</b>	
Option Public	
Option Declare	
Use "slCore"	
Dim thisUI As NotesUIDocument	
Dim thisNote As NotesDocument	
Sub SetCounts()	
Dim openView As NotesView	
Dim nav As NotesViewNavigator	
Dim dc As NotesDocumentCollection	
Dim keys(1) As String	
Set openView=DB.GetView("(users)")	
Call openView.Refresh()	
openView.Autoupdate=False	
Set nav=openView.CreateViewNav	
SetAdminValue "PersonDocCount",nav.Count	
Set openView=DB.GetView("dagroups")	

```

Call openView.Refresh()
openView.Autoupdate=False
Set nav=openView.CreateViewNav
SetAdminValue "GroupDocCount",nav.Count
Set openView=DB.GetView("(dauser)")
Call openView.Refresh()
openView.Autoupdate=False
Set nav=openView.CreateViewNav
SetAdminValue "DAUserCount",nav.Count
Set openView=DB.GetView("(mailuser)")
Call openView.Refresh()
openView.Autoupdate=False
Set nav=openView.CreateViewNav
SetAdminValue "MailUserCount",nav.Count
Set openView=DB.GetView("(valid)")
Call openView.Refresh()
openView.Autoupdate=False
Set nav=openView.CreateViewNav
SetAdminValue "ValidUserCount",nav.Count
keys(0)="all"
keys(1)="ccb"
Set openView=DB.GetView("(counts)")
Call openView.Refresh()
openView.Autoupdate=False
Set dc=openView.GetAllDocumentsByKey(keys,True)
SetAdminValue "ValidUserCountCCB",dc.Count
keys(1)="ccxtrue"
Set dc=openView.GetAllDocumentsByKey(keys,True)
SetAdminValue "ValidUserCountCCXActive",dc.Count
keys(1)="ccxfalse"
Set dc=openView.GetAllDocumentsByKey(keys,True)
SetAdminValue "ValidUserCountCCXInactive",dc.Count

```

End Sub

**(Form) fmWelcome**

```

Sub Postopen(Source As NotesUIDocument)
    QO
    PO
    Set thisUI=source
    Set thisNote=source.Document
    Dim adminVal As String
    adminVal=Cstr(GetAdminValue("ObservationsSecurity")(0))
    If adminVal<>"0" And adminVal<>"" Then thisNote.ShowWarning="y"
    Call source.Refresh
    Call source.RefreshHideFormulas

```

End Sub

**(Button) GetInfo**

```

Sub Click(Source As Button)
    On Error Goto errHandle
    Exit Sub
    Dim msgStr As String
    If db Is Nothing Then QO
    If ws Is Nothing Then PO
    Dim dialog As New NotesDocument(db)
    thisNote.GetInfoCancelled=""
    If GetAdminValue ("Name")(0)="" Then
        If Not ws.dialogbox("dlgUserInfo",t,t,f,f,f,"User Information",dialog,t,f,t) Then

```

```

        thisNote.GetInfoCancelled="true"
        Exit Sub
    End If
    If dialog.CompanyName(0)="" Then msgStr=msgStr+cr+tb+"- Company Name"
    If dialog.HCLAccount(0)="" Then msgStr=msgStr+cr+tb+"- HCL Account #"
    If msgStr<>"" Then
        MsgBox "Not all information was provided. All information is required."+cr+msgStr
        thisNote.GetInfoCancelled="true"
        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)
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

```

### Name:

Alias:  
 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:

### fmSidebar1

fmSidebar  
 07/04/2023 10:52:19 AM  
 [Not Assigned]  
 Document  
 No  
 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:  
 Is Scripted:

### SaveOptions

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



**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:

**Option**

Number  
General  
No  
No  
No  
[Not Assigned]  
Computed when composed  
No  
Disabled  
No  
1  
No

**Subform:****sfmSidebarOptions****JavaScript & HTML Code:**

[None]

**LotusScript Code:**

[None]

**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:

**fmSidebar2**

07/04/2023 10:52:17 AM  
[Not Assigned]  
Document  
No  
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:  
Is Scripted:

**SaveOptions**

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

**Field:**

Datatype:  
Number Format:  
Percentage (value \* 100)%:  
Parentheses on Negative Numbers:  
Punctuated at Thousands:

**Option**

Number  
General  
No  
No  
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
<b>Subform:</b>	<b>sfmSidebarOptions</b>
<b>JavaScript &amp; HTML Code:</b>	
[None]	
<b>LotusScript Code:</b>	
[None]	
<b>Name:</b>	<b>fmSidebar3</b>
Last Modification:	07/04/2023 10:52:17 AM
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;
<b>Subcomponents:</b>	
<b>Field:</b>	<b>SaveOptions</b>
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
<b>Field:</b>	<b>Option</b>
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:  
Subform:  
**JavaScript & HTML Code:**  
[None]  
**LotusScript Code:**  
[None]

No  
**sfmSidebarOptions**

**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:

**fmUser**  
User  
07/04/2023 10:52:17 AM  
[Not Assigned]  
Document  
"HCL Domino User Information"  
No  
No  
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:  
Is Scripted:

**LicenseTimestamp**  
Time  
07/04/2023  
[Not Assigned]  
Computed when composed  
No  
Disabled  
No  
@ThisValue  
No

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

**NABNOTECERTEXPIRY**  
Time  
07/04/2023  
[Not Assigned]  
Computed when composed  
No  
Disabled  
No  
@ThisValue  
No

**Action:**  
**Action:**  
**Action:**  
**Action:**  
Formula:  
**Action:**  
Formula:

**Spacer**  
**FormName**  
**Spacer**  
**Close**  
@Command([CloseWindow]);  
**Print**  
@Command([FilePrint]);

## JavaScript & HTML Code:

[None]

## 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  
07/04/2023 10:52:18 AM  
[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:  
Help Description:  
Field Type:  
Sign When Mailing/In Section:  
Encryption:  
Update Requires Editor Access:  
Formula:

### LDAPDA

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

Is Scripted:

No

### Field:

Datatype:  
Number Format:  
Percentage (value \* 100)%:

### ValidUserCountCCB

Number  
General  
No

Parentheses on Negative Numbers: Punctuated at Thousands: Help Description: Field Type: Sign When Mailing/In Section: Encryption: Update Requires Editor Access: Formula:	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: <b>Field:</b> 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:	No <b>ValidUserCountCCXActive</b> 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: <b>Field:</b> 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:	No <b>ValidUserCountCCXInactive</b> 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: <b>Field:</b> Datatype: Number Format: Percentage (value * 100)%: Parentheses on Negative Numbers: Punctuated at Thousands: Help Description: Field Type: Sign When Mailing/In Section:	No <b>ValidUserCount</b> Number General No No No [Not Assigned] Computed when composed No

Encryption:	Disabled
Update Requires Editor Access:	No
Formula:	ccb := ValidUserCountCCB;ccx := ValidUserCountCCXActive; valid := @If(ccb > 0 & ccx > -1; ccb + ccx; ValiduserCount); @If(@IsError(valid); "Unknown"; valid = -10101; "Error in Calculation"; valid > -1; @Text(valid); "Calculating");
Is Scripted:	No
<b>Field:</b>	<b>ValidUserCountCCBDisplay</b>
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 for display
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Formula:	@If(ValidUserCountCCB = 0; ValidUserCount; ValidUserCountCCB);
Is Scripted:	No
Hide When:	ValidUserCountCCB = "";
<b>Field:</b>	<b>ValidUserCountCCXActiveDisplay</b>
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 for display
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Formula:	ValidUserCountCCXActive
Is Scripted:	No
Hide When:	ValidUserCountCCXActive = "";
<b>Field:</b>	<b>ValidUserCountCCXInactiveDisplay</b>
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 for display
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No

Formula:	ValidUserCountCCXInactive
Is Scripted:	No
Hide When:	ValidUserCountCCXInactive = ""
<b>Field:</b>	<b>ValidUserCountGuest</b>
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
<b>Field:</b>	<b>MailUserCount</b>
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); "< Error Retrieving Value >"; val);
Is Scripted:	No
<b>Field:</b>	<b>DAUserCount</b>
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); "< Error Retrieving Value >"; val);
Is Scripted:	No
<b>Field:</b>	<b>PersonDocCount</b>
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); "< Error Retrieving Value >"; val);
Is Scripted:	No
<b>Field:</b>	<b>CompanyName</b>
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
<b>Field:</b>	<b>HCLAccount</b>
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
<b>Field:</b>	<b>Name</b>
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
Hide When:	Name = "";
<b>Field:</b>	<b>Email</b>



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
Hide When:	Email = "";
<b>Field:</b>	<b>Phone</b>
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
Hide When:	Phone = "";
<b>Hotspot</b>	
Type:	Action Hotspot
Caption:	Submit Report
Is Scripted:	Yes, see below
<b>Hotspot</b>	
Type:	Action Hotspot
Caption:	Print Report
Formula:	@Command([FilePrint]);
<b>Hotspot</b>	
Type:	Action Hotspot
Caption:	Return to Scan Results
Formula:	@SetTargetFrame("Inner"); @Command([OpenFrameset]; "fsInner");
<b>JavaScript &amp; HTML Code:</b>	
[None]	
<b>LotusScript Code:</b>	
(Globals)	
Option Public	
Option Declare	
Use "slCore"	
<b>(Hotspot) Submit Report</b>	
Sub Click(Source As Button)	
' collect additional information	
Dim encString As String	
Dim tmpString As String	
Dim signature As String	
Dim bodyString As String	
Dim length As Integer	

```

Dim url As String
Dim hexValue As String
Dim x As Integer
Dim num As String
QQ
PO
Const nl="%0D%0A%0D%0A"
If MsgBox("I acknowledge that I am not a US Federal Government employee or agency, nor am I submitting
url="mailto:hcl_domino_license_submission@pnp-hcl.com?subject=HCL+DLAU+License+Report&body="
' current Digital Signature
signature=GetAdminValue("Signature")(0)
signature=Replace(signature,"=", "Y2hyaXN0aW5l")
num=Left(signature,1)
signature=Right(signature,Len(signature)-1)
signature=Flip(signature)
If num="2" Then num="D"
If num="3" Then num="F"
If num="4" Then num="J"
If num="5" Then num="M"
If num="6" Then num="P"
signature=num+signature
bodyString=bodyString+
' HCL Account #
tmpString=GetAdminValue("HCLAccount")(0)
encString=|"HCLAccount:"|+tmpString+|"|
bodyString="HCL Account: "|+tmpString+nl
' Company Name
tmpString=GetAdminValue("CompanyName")(0)
encString=encString+|,|+nl+|"Company:"|+tmpString+|"|
bodyString=bodyString+"Company: "|+tmpString+nl
' Username (if provided)
tmpString=GetAdminValue("Name")(0)
If tmpString<>"" Then
    encString=encString+|,|+nl+|"UserName:"|+tmpString+|"|
    bodyString=bodyString+"User Name: "|+tmpString+nl
End If
' Email (if provided)
tmpString=GetAdminValue("Email")(0)
If tmpString<>"" Then
    encString=encString+|,|+nl+|"Email:"|+tmpString+|"|
    bodyString=bodyString+"Email: "|+tmpString+nl
End If
' Phone # (if provided)
tmpString=GetAdminValue("Phone")(0)
If tmpString<>"" Then
    encString=encString+|,|+nl+|"Phone:"|+tmpString+|"|
    bodyString=bodyString+"Phone: "|+tmpString+nl
End If
' get rest of values for email
bodyString=bodyString+"Authorized User Count: "+Cstr(note.ValidUserCount(0))+nl
bodyString=bodyString+"CCB License Count: "+Cstr(note.ValidUserCountCCB(0))+nl
bodyString=bodyString+"CCX License Count: "+Cstr(note.ValidUserCountCCXActive(0))+nl
bodyString=bodyString+"CCX Inactive Count: "+Cstr(note.ValidUserCountCCXInactive(0))+nl
bodyString=bodyString+"Known Guests Count: "+Cstr(note.ValidUserCountGuest(0))+nl
bodyString=bodyString+"Mail User Count: "+Cstr(note.MailUserCount(0))+nl
bodyString=bodyString+"Deny Access User Count: "+Cstr(note.DAUserCount(0))+nl

```

```

bodyString=bodyString+"Person Document Count: "+Cstr(note.PersonDocCount(0))+nl
bodyString=bodyString+nl+"Note: Do Not modify the data below!"+nl
' replace all spaces with %20 (ascii value for a space)
bodyString=Replace(bodyString," ","%20")
' encrypt the above information
encString=EncodeBase64SL(encString)
encString=Flip(encString)
hexValue=Hex$(Len(encString))
tmpString=Cstr(Len(hexValue))
' now create the data that will be sent with the URL
encString=signature+encString+hexValue+tmpString
encString=Replace(encString,"=", "RGV2b255")
Call ws.URLOpen(url+bodyString+encstring)
End Sub
Function Flip(Byval inputText As String) As String
    Dim reversedText As String
    Dim i As Integer
    For i = Len(inputText) To 1 Step -1
        reversedText = reversedText & Mid(inputText, i, 1)
    Next i
    Flip = reversedText

```

## View Information

### **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:

### **Deny Access Groups**

dagroups  
 07/04/2023 10:52:16 AM  
 [Not Assigned]  
 No  
 Disabled  
 Expanded  
 None  
 Automatic  
 Never  
 All Users  
 No  
 SELECT Form = "Group";

### **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:

1  
 Lookup  
 ListName  
 20 Characters  
  
 Yes  
 No  
 No  
 General  
 No  
 No  
 No  
 07/04/2023 10:53:04 AM  
 Ascending  
 Left  
 None

Hide Detail Values:	No
<b>Column Number:</b>	<b>2</b>
Column Title:	Group Name
Column Formula:	@Name([Abbreviate]; 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:	07/04/2023 10:53:04 AM
Sorting:	Ascending
Justification:	Left
Totals:	None
Hide Detail Values:	No
<b>Column Number:</b>	<b>3</b>
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:	07/04/2023 10:53:04 AM
Sorting:	Ascending
Justification:	Left
Totals:	None
Hide Detail Values:	No
<b>Column Number:</b>	<b>4</b>
Column Title:	# Members
Column Formula:	@If(@IsAvailable(MemberQuan tity); MemberQuantity; 0);
Column Width:	8 Characters
List Separator:	
Hidden:	Yes
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:	07/04/2023 10:53:04 AM
Sorting:	Ascending
Justification:	Right
Totals:	None
Hide Detail Values:	No
<b>Column Number:</b>	<b>5</b>

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:	07/04/2023 10:53:04 AM
Sorting:	Ascending
Justification:	Right
Totals:	None
Hide Detail Values:	No
<b>Column Number:</b>	<b>6</b>
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:	07/04/2023 10:53:04 AM
Sorting:	No
Justification:	Left
Totals:	None
Hide Detail Values:	No
<b>Column Number:</b>	<b>7</b>
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:	07/04/2023 10:53:04 AM
Sorting:	Ascending
Justification:	Left
Totals:	None
Hide Detail Values:	No
<b>JavaScript &amp; HTML Code:</b>	
	[None]
<b>LotusScript Code:</b>	
	[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:

**Observations**

Observations  
07/04/2023 10:52:18 AM  
[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

07/04/2023 10:53 AM

Ascending, Categorized

Left

None

No

**2**

Date of Scan

@Created

14 Characters

No

No

No

General

No

No

No

07/04/2023 10:53 AM

Descending

Left

None

Hide Detail Values:	No
<b>Column Number:</b>	<b>3</b>
Column Title:	Administration Server
Column Formula:	AdminServer
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:	07/04/2023 10:53:04 AM
Sorting:	Ascending
Justification:	Left
Totals:	None
Hide Detail Values:	No
<b>Column Number:</b>	<b>4</b>
Column Title:	# Observations
Column Formula:	warn := @If(ObservationsWarning = ""; 0; ObservationsWarning);sec := @If(ObservationsSecurity = ""; 0; ObservationsSecurity);info := @If(ObservationsInformation = ""; 0; ObservationsInformation); warn + sec + info;
Column Width:	9 Characters
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:	07/04/2023 10:53:04 AM
Sorting:	Ascending
Justification:	Right
Totals:	None
Hide Detail Values:	No
<b>Column Number:</b>	<b>5</b>
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:	07/04/2023 10:53:04 AM
Sorting:	Ascending

Justification:	Right
Totals:	None
Hide Detail Values:	No
<b>Column Number:</b>	<b>6</b>
Column Title:	# Warnings
Column Formula:	ObservationsWarning
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:	07/04/2023 10:53:04 AM
Sorting:	Ascending
Justification:	Right
Totals:	None
Hide Detail Values:	No
<b>Column Number:</b>	<b>7</b>
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:	07/04/2023 10:53:04 AM
Sorting:	Ascending
Justification:	Right
Totals:	None
Hide Detail Values:	No
<b>JavaScript &amp; HTML Code:</b>	
[None]	
<b>LotusScript Code:</b>	
[None]	
<b>Name:</b>	<b>Scan History</b>
Alias:	History
Last Modification:	07/04/2023 10:52:18 AM
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:	07/04/2023 10:53 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:	07/04/2023 10:53:04 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:	07/04/2023 10:53:04 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:	07/04/2023 10:53:04 AM
Sorting:	Ascending
Justification:	Right
Totals:	None
Hide Detail Values:	No
<b>Column Number:</b>	<b>5</b>
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:	07/04/2023 10:53:04 AM
Sorting:	Ascending
Justification:	Right
Totals:	None
Hide Detail Values:	No
<b>Column Number:</b>	<b>6</b>
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:	07/04/2023 10:53:04 AM
Sorting:	Ascending
Justification:	Right
Totals:	None
Hide Detail Values:	No
<b>JavaScript &amp; HTML Code:</b>	
[None]	
<b>LotusScript Code:</b>	
[None]	
<b>Name:</b>	<b>User Information</b>
Alias:	UserInfo
Last Modification:	07/04/2023 10:52:19 AM

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  
Collapsed  
None  
Automatic  
Never  
All Users  
No  
SELECT Form = "User";

### 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  
[Not Assigned]  
1  
6 Characters  
  
No  
No  
No  
General  
No  
No  
No  
07/04/2023 10:53:04 AM  
No  
Right  
Total  
Yes

#### Column Number:

Column Title:  
Column Formula:

2  
[Not Assigned]  
certExpired :=  
CertificateExpired;httpPWSet :=  
HTTPPasswordSet;da :=  
DAUser;extNAB :=  
ExternalNAB;extORG :=  
ExternalORG;aclLevel :=  
@Word(EntitlementList; "~"; 2);  
licenseType :=  
@If(@IsError(aclLevel); "<  
Unknown >"; (aclLevel =  
"Author" | aclLevel = "Depositor"  
| aclLevel = "Reader" | aclLevel =  
"No Access" | aclLevel = "") &  
(extNAB + extORG != ""));  
"CCX"; "CCB Users");partial :=  
@If(PartialDAUser = ""; "";  
"\Partial Server Access");  
category := @If(KnownGuest =  
"true"; "Known Guest";  
licenseType = "CCX" &  
ActiveUser = "" & da = ""; "CCX  
Inactive"; licenseType = "CCX";  
"CCX Active"; UserType =  
"Unknown / Unlicensed";  
UserType; HTTPSecure =  
"false" & httpPWSet = "true";

Column Width:	"Authorized"; httpPWSet = "true"
List Separator:	& da != "" & HTTPSecure =
Hidden:	"true" & certExpired = "true";
Responses Only:	"Invalid"; da != ""; "Invalid";
Icon:	httpPWSet = "true";
Number Format:	"Authorized"; certExpired =
Percentage (value * 100)%:	"true"; "Invalid"; "Authorized") + "
Parentheses on Negative Numbers:	Users";@If(category =
Punctuated at Thousands:	"Authorized Users"; licenseType
Time Format:	+ @If(partial = ""; "\\Full Server
Sorting:	Access"; partial); category);
Justification:	8 Characters
Totals:	No
Hide Detail Values:	No
<b>Column Number:</b>	No
Column Title:	General
Column Formula:	No
Column Width:	No
List Separator:	No
Hidden:	No
Responses Only:	07/04/2023 10:53:04 AM
Icon:	Ascending, Categorized
Number Format:	Left
Percentage (value * 100)%:	None
Parentheses on Negative Numbers:	No
Punctuated at Thousands:	3
Time Format:	Last Accessed hidden
Sorting:	@If(LicenseTimestamp = "";
Justification:	@Date(1; 1; 1);
Totals:	LicenseTimestamp);
Hide Detail Values:	10 Characters
<b>Column Number:</b>	Yes
Column Title:	No
Column Formula:	No
Column Width:	General
List Separator:	No
Hidden:	No
Responses Only:	No
Icon:	No
Number Format:	07/2023
Percentage (value * 100)%:	Descending
Parentheses on Negative Numbers:	Left
Punctuated at Thousands:	None
Time Format:	No
Sorting:	No
Justification:	4
Totals:	Last Accessed
Hide Detail Values:	@If(LicenseTimestamp = ""; " ";
<b>Column Number:</b>	LicenseTimestamp);
Column Title:	10 Characters
Column Formula:	
Column Width:	No
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:	07/2023
Sorting:	No
Justification:	Left
Totals:	None
Hide Detail Values:	No
<b>Column Number:</b>	<b>5</b>
Column Title:	Full Name
Column Formula:	@Name([Abbreviate]; 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:	07/04/2023 10:53:04 AM
Sorting:	Ascending
Justification:	Left
Totals:	None
Hide Detail Values:	No
<b>Column Number:</b>	<b>6</b>
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:	07/04/2023 10:53:04 AM
Sorting:	No
Justification:	Left
Totals:	None
Hide Detail Values:	No
<b>Column Number:</b>	<b>7</b>
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:	07/2023
Sorting:	Ascending

Justification:	Left
Totals:	None
Hide Detail Values:	No
<b>Column Number:</b>	<b>8</b>
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:	07/2023
Sorting:	Ascending
Justification:	Left
Totals:	None
Hide Detail Values:	No
<b>Column Number:</b>	<b>9</b>
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:	07/04/2023
Sorting:	No
Justification:	Left
Totals:	None
Hide Detail Values:	No
<b>Column Number:</b>	<b>10</b>
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:	07/04/2023 10:53:04 AM
Sorting:	Ascending
Justification:	Left
Totals:	None
Hide Detail Values:	No
<b>JavaScript &amp; HTML Code:</b>	

[None]

## LotusScript Code:

### (Action) Export

```
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(6) As String
    Dim selection As String
    Dim folder As String
    Set db=session.CurrentDatabase
    options(0)="All User Documents"
    options(1)="CCB Users"
    options(2)="CCX Active Users"
    options(3)="CCX Inactive Users"
    options(4)="Invalid Users"
    options(5)="Known Guest Users"
    options(6)="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")
        Call view.Refresh
        view.AutoUpdate=False
        Set dc=view.GetAllDocumentsByKey("User")
    Else
        Set view=db.GetView("export")
        Call view.Refresh
        view.AutoUpdate=False
        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:

## (config)

config  
 07/04/2023 10:52:18 AM  
 [Not Assigned]  
 No  
 Disabled  
 Expanded  
 None  
 Automatic  
 Never  
 All Users  
 No



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

SELECT Form = "config";  
  
**1**  
#  
Form  
10 Characters  
  
No  
No  
No  
General  
No  
No  
No  
07/04/2023 10:53:04 AM  
Ascending  
Left  
None  
No

**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:

**(counts)**  
counts  
07/04/2023 10:52:18 AM  
[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  
= "Valid" & UserType !=  
"Unknown / Unlicensed";

**Subcomponents:**  
  **Column Number:**  
    Column Title:  
    Column Formula:

**1**  
all  
"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:	07/04/2023 10:53:04 AM
Sorting:	Ascending
Justification:	Left
Totals:	None
Hide Detail Values:	No
<b>Column Number:</b>	<b>2</b>
Column Title:	[Not Assigned]
Column Formula:	certExpired := CertificateExpired;httpPWSet := HTTPPasswordSet;da := DAUser;extNAB := ExternalNAB;extORG := ExternalORG;aclLevel := @Word(EntitlementList; "~"; 2); licenseType := @if(@IsError(aclLevel); "< Unknown >"; (aclLevel = "Author"   aclLevel = "Depositor"   aclLevel = "Reader"   aclLevel = "No Access"   aclLevel = "")) & (extNAB + extORG != ""); "CCX"; "CCB");category := @if(KnownGuest = "true"; "Known Guest"; licenseType = "CCX" & ActiveUser = "" & da = ""; "CCX Inactive"; licenseType = "CCX"; "CCX Active"; 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"); @LowerCase(@If(category = "Authorized"; licenseType; category));
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:	07/04/2023 10:53:04 AM
Sorting:	Ascending
Justification:	Left
Totals:	None
Hide Detail Values:	No
<b>JavaScript &amp; HTML Code:</b>	
[None]	
<b>LotusScript Code:</b>	
[None]	
<b>Name:</b>	<b>(dauser)</b>
Alias:	dauser
Last Modification:	07/04/2023 10:52:18 AM
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" & DAUser != "" & UserType != "Unkown / Unlicensed";
<b>Subcomponents:</b>	
<b>Column Number:</b>	<b>1</b>
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:	07/04/2023 10:53:04 AM
Sorting:	Ascending
Justification:	Left
Totals:	None
Hide Detail Values:	No
<b>JavaScript &amp; HTML Code:</b>	
[None]	
<b>LotusScript Code:</b>	
[None]	
<b>Name:</b>	<b>(export)</b>
Alias:	export

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:

07/04/2023 10:52:18 AM  
[Not Assigned]  
No  
Disabled  
Collapsed  
None  
Automatic  
Never  
All Users  
No  
SELECT Form = "User";

**Subcomponents:**

**Column Number:**  
Column Title:  
Column Formula:

**1**  
Lookup  
certExpired :=  
CertificateExpired;httpPWSet :=  
HTTPPasswordSet;da :=  
DAUser;extNAB :=  
ExternalNAB;extORG :=  
ExternalORG;aclLevel :=  
@Word(EntitlementList; "~"; 2);  
licenseType :=  
@If(@IsError(aclLevel); "<  
Unknown >"; (aclLevel =  
"Author" | aclLevel = "Depositor"  
| aclLevel = "Reader" | aclLevel =  
"No Access" | aclLevel = "")) &  
(extNAB + extORG != "");  
"CCX"; "CCB Users");partial :=  
@If(PartialDAUser = ""; "";  
"\Partial Server Access");  
category := @If(KnownGuest =  
"true"; "Known Guest";  
licenseType = "CCX" &  
ActiveUser = "" & da = ""; "CCX  
Inactive"; licenseType = "CCX";  
"CCX Active"; 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"; licenseType  
+ @If(partial = ""; "\Full Server  
Access"; partial); category);  
16 Characters

Column Width:  
List Separator:  
Hidden:  
Responses Only:  
Icon:

No  
No  
No

Number Format:	General
Percentage (value * 100)%:	No
Parentheses on Negative Numbers:	No
Punctuated at Thousands:	No
Time Format:	07/04/2023 10:53:04 AM
Sorting:	Ascending
Justification:	Left
Totals:	None
Hide Detail Values:	No
<b>JavaScript &amp; HTML Code:</b>	
[None]	
<b>LotusScript Code:</b>	
[None]	
<b>Name:</b>	<b>(groupmembers)</b>
Alias:	members
Last Modification:	07/04/2023 10:52:18 AM
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" & Members != "";
<b>Subcomponents:</b>	
<b>Column Number:</b>	<b>1</b>
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:	07/04/2023 10:53:04 AM
Sorting:	Ascending
Justification:	Left
Totals:	None
Hide Detail Values:	No
<b>Column Number:</b>	<b>2</b>
Column Title:	Server
Column Formula:	@LowerCase(@Trim(Servers));
Column Width:	40 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:	07/04/2023 10:53:04 AM
Sorting:	Ascending
Justification:	Left
Totals:	None
Hide Detail Values:	No
<b>JavaScript &amp; HTML Code:</b>	
[None]	
<b>LotusScript Code:</b>	
[None]	
<b>Name:</b>	<b>(invalid)</b>
Alias:	invalid
Last Modification:	07/04/2023 10:52:18 AM
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 = "" & UserType != "Unknown / Unlicensed";
<b>Subcomponents:</b>	
<b>Column Number:</b>	<b>1</b>
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:	07/04/2023 10:53:04 AM

Sorting:  
Justification:  
Totals:  
Hide Detail Values:  
**JavaScript & HTML Code:**  
[None]  
**LotusScript Code:**  
[None]

Ascending  
Left  
None  
No

**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:

**(mailuser)**

mailuser  
07/04/2023 10:52:18 AM  
[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  
= "Valid" & UserType !=  
"Unknown / Unlicensed" &  
MailUser != "";

**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  
#  
"all"  
10 Characters  
  
No  
No  
No  
General  
No  
No  
No  
07/04/2023 10:53:04 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:

**(parentgroups)**

parents

07/04/2023 10:52:18 AM

[Not Assigned]

No

Disabled

Expanded

None

Automatic

Never

All Users

No

SELECT Form = "Group" &amp;

TopLevel != "";

**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

07/04/2023 10:53:04 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:

**(possibleusers)**

possibleusers

07/04/2023 10:52:18 AM

[Not Assigned]

No

Disabled

Expanded

None

Automatic

Never

All Users



Time Relative Formulas:  
Selection Formula:

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  
07/04/2023 10:53:04 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  
07/04/2023 10:53:04 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:

(subgroupsunprocessed)  
sgunprocessed  
07/04/2023 10:52:18 AM  
[Not Assigned]  
No  
Disabled  
Expanded  
None  
Automatic  
Never

View May Be Used By:  
Time Relative Formulas:  
Selection Formula:

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  
07/04/2023 10:53:04 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  
07/04/2023 10:52:18 AM  
[Not Assigned]  
No  
Disabled  
Expanded  
None  
Automatic  
Never  
All Users  
No  
SELECT Form = "Group" &  
TopLevel = "";

### Subcomponents:

#### Column Number:

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

1  
Lookup  
ListName  
20 Characters  
  
No  
No  
No  
General  
No  
No

Punctuated at Thousands:	No
Time Format:	07/04/2023 10:53:04 AM
Sorting:	Ascending
Justification:	Left
Totals:	None
Hide Detail Values:	No
<b>JavaScript &amp; HTML Code:</b>	
[None]	
<b>LotusScript Code:</b>	
[None]	

<b>Name:</b>	<b>(users)</b>
Alias:	users
Last Modification:	07/04/2023 10:52:18 AM
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";
<b>Subcomponents:</b>	
<b>Column Number:</b>	<b>1</b>
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:	07/04/2023 10:53:04 AM
Sorting:	Ascending
Justification:	Left
Totals:	None
Hide Detail Values:	No
<b>Column Number:</b>	<b>2</b>
Column Title:	DA User
Column Formula:	DAUSER
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:	07/04/2023 10:53:04 AM
Sorting:	Ascending
Justification:	Left
Totals:	None
Hide Detail Values:	No
<b>JavaScript &amp; HTML Code:</b>	
[None]	
<b>LotusScript Code:</b>	
[None]	
<b>Name:</b>	<b>(valid)</b>
Alias:	valid
Last Modification:	07/04/2023 10:52:18 AM
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";
<b>Subcomponents:</b>	
<b>Column Number:</b>	<b>1</b>
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:	07/04/2023 10:53:04 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:

**JavaScript & HTML Code:**

[None]

**LotusScript Code:**

[None]

**2**

License Type

extNAB := ExternalNAB;extORG

:= ExternalORG;aclLevel :=

@Word(EntitlementList; "~"; 2);

type := @If(@IsError(aclLevel);

"&lt; Unknown &gt;"; KnownGuest !=

"" | aclLevel = ""; "Guest";

(aclLevel = "Author" | aclLevel =

"" ) &amp; (extNAB + extORG != "");

"CCX"; "CCB");type + @If(type =

"CCX"; @If(ActiveUser = "";

"false"; ActiveUser); "");

10 Characters

No

No

No

General

No

No

No

07/04/2023 10:53:04 AM

Ascending

Left

None

No

**Name:**

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:

**(\$All)**

07/04/2023 10:52:18 AM

[Not Assigned]

No

Disabled

Expanded

None

Automatic

Never

All Users

No

SELECT @All

**Subcomponents:****Column Number:**

Column Title:

Column Formula:

Column Width:

List Separator:

Hidden:

Responses Only:

Icon:

Number Format:

**1**

#

1

10 Characters

No

No

No

General

Percentage (value * 100)%:	No
Parentheses on Negative Numbers:	No
Punctuated at Thousands:	No
Time Format:	07/04/2023 10:53:04 AM
Sorting:	No
Justification:	Left
Totals:	Total
Hide Detail Values:	Yes
<b>Column Number:</b>	<b>2</b>
Column Title:	Form
Column Formula:	@If(Form = "config"; "1. Scan Log"; Form);
	10 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:	07/04/2023 10:53:04 AM
Sorting:	Ascending, Categorized
Justification:	Left
Totals:	None
Hide Detail Values:	No
<b>Column Number:</b>	<b>3</b>
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:	07/04/2023 10:53:04 AM
Sorting:	No
Justification:	Left
Totals:	None
Hide Detail Values:	No
<b>JavaScript &amp; HTML Code:</b>	
[None]	
<b>LotusScript Code:</b>	
[None]	

## Subform Information

<b>Name:</b>	<b>sfmlInfoOverlays</b>
Last Modification:	07/04/2023 10:52:16 AM
Include in Insert Subform list:	Yes
Include in New Form dialog:	No
<b>Subcomponents:</b>	

**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:  
Default Value Formula:

**nl**

Text  
[Not Assigned]  
Editable  
No  
Disabled  
No  
@NewLine  
No

**AU**

Text  
[Not Assigned]  
Editable  
No  
Disabled  
No  
"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;

**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:	No
<b>Field:</b>	<b>DAG</b>
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 Deny Access Groups that are listed on the HCL Domino server document(s)."
Is Scripted:	No
<b>Field:</b>	<b>DAU</b>
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 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:	No
<b>Field:</b>	<b>AS</b>
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 HCL Domino Administration server that was the first HCL Domino server in your environment. Most, if not all, server configuration and



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

User details are stored on this HCL Domino server."

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 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."

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"

No

**ADD**

Text

[Not Assigned]

Editable

No

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

Encryption:	Disabled
Update Requires Editor Access:	No
Default Value Formula:	"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:	No
<b>Field:</b>	<b>OB</b>
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 a list of findings that 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";
Is Scripted:	No
<b>Field:</b>	<b>MU</b>
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 total number of Mail Users discovered in your HCL Domino server environment"
Is Scripted:	No
<b>Field:</b>	<b>TitleDisp</b>
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:	info := ShowInfo;options := "AU" : "PD" : "DAG" : "DAU" : "AS" : "DSE" : "ET" : "DA" : "ADD" :

Is Scripted:	"OB" : "MU";visual :=
<b>Hotspot</b>	"Authorized Users" : "Person
Type:	Documents" : "Deny Access
Caption:	Groups" : "Deny Access Users" :
Formula:	"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]);
	@Command([RefreshHideForm
	ulas]);
<b>Field:</b>	<b>InfoDisp</b>
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:	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
Is Scripted:	
<b>JavaScript &amp; HTML Code:</b>	
[None]	
<b>LotusScript Code:</b>	
[None]	
<b>Name:</b>	<b>sfmSidebarOptions</b>
Last Modification:	07/04/2023 10:52:17 AM
Include in Insert Subform list:	Yes
Include in New Form dialog:	No
<b>Subcomponents:</b>	
<b>Field:</b>	<b>ver</b>
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:  
Formula:  
Is Scripted:  
**Graphic Button**  
Formula:

No  
@TextToNumber(@Version);  
No

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

**Hotspot**  
Type:  
Caption:  
Formula:

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

**Graphic Button**  
Formula:

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

**Hotspot**  
Type:  
Caption:  
Formula:

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

**Graphic Button**  
Formula:

FIELD Option := 3;  
@Command([RefreshHideFormulas]);  
@SetTargetFrame("Main");  
@PostedCommand([OpenView]; "dagroups");

**Hotspot**  
Type:  
Caption:  
Formula:

Action Hotspot  
Deny Access Groups  
FIELD Option := 3;  
@Command([RefreshHideFormulas]);  
@SetTargetFrame("Main");  
@PostedCommand([OpenView]; "dagroups");

**Graphic Button**  
Formula:

FIELD Option := 5;  
@Command([RefreshHideFormulas]);  
@SetTargetFrame("Main");

**Hotspot**

Type:  
Caption:  
Formula:

```
@PostedCommand([OpenView]
; "Observations");
```

Action Hotspot  
Observations  
FIELD Option := 5;  
@Command([RefreshHideForm  
ulas]);  
@SetTargetFrame("Main");  
@PostedCommand([OpenView]  
; "Observations");

**Graphic Button**

Formula:

FIELD Option := 4;  
@Command([RefreshHideForm  
ulas]);  
@SetTargetFrame("Main");  
@PostedCommand([OpenView]  
; "History");

**Hotspot**

Type:  
Caption:  
Formula:

Action Hotspot  
Previous Results  
FIELD Option := 4;  
@Command([RefreshHideForm  
ulas]);  
@SetTargetFrame("Main");  
@PostedCommand([OpenView]  
; "History");

**Graphic Button**

Is Scripted:

Yes, see below

**Hotspot**

Type:  
Caption:  
Is Scripted:

Action Hotspot  
Scan Log  
Yes, see below

**Graphic Button**

Formula:

```
@SetTargetFrame("Inner");
@PostedCommand([OpenFram
eset]; "fsLaunch");
```

**Hotspot**

Type:  
Caption:  
Formula:

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

**JavaScript & HTML Code:**

[None]

**LotusScript Code:****(Hotspot) 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
    MsgBox "No Scan Log could be located. Please rerun the Wizard to have a Scan Log created."
Exit Sub
End If
logWS.CurrentDocument.Document.Option=6
Call logWS.CurrentDocument.RefreshHideFormulas
Dim logURL As String
logURL=logNote.NotesURL
Call logSession.SetEnvironmentVar("DLAULogURL",logNote.NotesURL)
Call logWS.SetTargetFrame("Main")
Call logWS.OpenFrameSet("fsLog")

```

## **Code Library Information**

### **Name:**

**slCore**

Last Modification:

07/04/2023 10:52:17 AM

### **LotusScript Code:**

```

Option Public
Option Declare
Use "slUtilities"
Dim dadb As NotesDatabase
Dim dirArray() As Variant
Dim replDArray() As String
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
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
```

' these are to allow us to check for duplicate Person Names faster than using a View lookup. This set of arrays

```
Dim allNames1 As Variant
Dim allNames2 As Variant
Dim allNames3 As Variant
Dim allNames4 As Variant
Dim allNames5 As Variant
Dim allNames6 As Variant
Dim allNames7 As Variant
Dim allNames8 As Variant
Dim allNames9 As Variant
Dim allNames10 As Variant
Dim allNames11 As Variant
Dim allNames12 As Variant
Dim allNames13 As Variant
Dim allNames14 As Variant
Dim allNames15 As Variant
Dim allNames16 As Variant
Dim allNames17 As Variant
Dim allNames18 As Variant
Dim allNames19 As Variant
Dim allNames20 As Variant
Dim allNames21 As Variant
Dim allNames22 As Variant
Dim allNames23 As Variant
Dim allNames24 As Variant
Dim allNames25 As Variant
Dim allNames26 As Variant
Dim allNames27 As Variant
Dim allNames28 As Variant
Dim allNames29 As Variant
Dim allNames30 As Variant
Dim allNames31 As Variant
Dim allNames32 As Variant
Dim allNames33 As Variant
Dim allNames34 As Variant
Dim allNames35 As Variant
Dim allNames36 As Variant
Dim allNames37 As Variant
Dim allNames38 As Variant
Dim allNames39 As Variant
Dim allNames40 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
    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
' check for the License Tracking db on the accessible servers
LogStep norm,CStr(Now)+tb+|In "CheckServerSettings" - checking for License Tracking apps|
CheckLicenseTrackingSettings
' Question 2: Extenal 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
    %rem
    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",""
    End If
    %end rem
    SetAdminValue "AskExternal","true"
End If
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=Err
    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
        Else 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
        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,t)
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 from the NAB
    LogStep norm,CStr(Now)+tb+|The Deny Access Group named "|+daGroups(x)+|" appears to be missing from the NAB
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 no values in the "Members" field. Just save the DA Group document
        daNote.Members=""
        Call daNote.Save(t,t,t)
        Observation 1007,|The Deny Access Group named "|+daGroups(x)+|" has no Members listed in the NAB
        LogStep norm,CStr(Now)+tb+|The Deny Access Group named "|+daGroups(x)+|" has no Members listed in the NAB
    ElseIf item.Text="" Then
        ' the Group in the NAB has no values in the "Members" field. Just save the DA Group document
        daNote.Members=""
        Call daNote.Save(t,t,t)
        Observation 1007,|The Deny Access Group named "|+daGroups(x)+|" has no Members listed in the NAB
        LogStep norm,CStr(Now)+tb+|The Deny Access Group named "|+daGroups(x)+|" has no Members listed in the NAB
    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 y
    End For
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=U
        If subGroups(0)="" Then daNote.SubgroupQuantity=0 Else daNote.SubgroupQuantity=U
        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
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 th
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 basedon 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 th
   - 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 tha

```

```

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
    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 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
        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 f
                    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
            daNote.Processed="true"
            Call daNote.Save(t,t,t)
        Else
            daNote.Processed="true"
            Call daNote.Save(t,t,t)
        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
If IsArray(session.Addressbooks) Then
    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)
                    Exit ForAll
                End If
            End If
        End If
    End ForAll
End If
' 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
LogStep norm,CStr(Now)+tb+|In "GetAdminServerInfo" - svrName = |+adminServer
' 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
ReDim replIDArray(0)
replIDArray(0)=nab.Replicaid
' 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
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
    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 l

```

```

        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 remaining servers
            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 field
    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 "+svrDisabledList(x)
            LogStep norm,CStr(Now)+tb+"|The License Tracking task is not running on the HCL Domino server "+svrDisabledList(x)
        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 Error are: "+errString
    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("$UpdatedBy").Values(UBound(viewNote.Getfirstitem($UpdatedBy).Values)))
    If valCheck.Organization<>currentUser.Organization Then Exit Function
    CheckDesign=t
errHandle:
    If Err<>0 Then
        errString=Error
        errNum=Err
        errLine=Err
        ReportError "There was an Error checking the signer of this HCL Notes application design (CheckDesign)"
    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 access
    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("AccessLevel")
    Exit Function
exitNoRole:
    MsgBox |You do not have the proper access to the HCL Domino Directory on the server: |+GetAdminValue("DenyLists")
    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 exiting
    Exit Function
End Function

```

```

Function EncodeBase64SL (StrIn As String) As String
    On Error GoTo errHandle
    Dim stream As NotesStream
    Dim tmpDoc As NotesDocument
    Dim body As NotesMIMEEntity
    If db Is Nothing Then QO
    Set stream=session.CreateStream
    Call stream.WriteText(StrIn)
    Set db = session.CurrentDatabase
    Set tmpDoc = db.CreateDocument
    Set body = tmpDoc.CreateMIMEEntity
    Call body.SetContentFromText (stream, "", ENC_NONE)
    Call body.EncodeContent (ENC_BASE64)
    EncodeBase64SL = Replace(Replace(body.ContentAsText, Chr(13),""), Chr(10),"")
errHandle:
    Call stream.Close
    Set tmpdoc = Nothing
    If Err<>0 Then MsgBox "error while encoding data"+Chr(10)+"error: "+Error+Chr(10)+"line: "+Cstr(Erl)+"", #
    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
    Dim isReader As Boolean
    Dim splitVar As Variant
    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)
        isReader=t
        For x=0 To UBound(valArray)
            splitVar=Split(valArray(x),"~")
            If LCase(splitVar(1))="manager" Or LCase(splitVar(1))="designer" Or LCase(splitVar(1))="editor" Then
                isReader=f
            End If
        Next
        If isReader Then userNote.KnownGuest="true"
        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("($Servers)")
    If svrView Is Nothing Then
        errString=|A View is missing from your Domino Directory - "($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 Trim(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"
                End If
            End If
        End If
        Set entry = nav.GetNextDocument(entry)
    End While
    Exit Function
End Function

```

```

        availCount=availCount+1
        online=t
    Else
        serverList(count)=serverList(count)+"|Offline"
        Observation 1001,|The HCL Domino Server "|+svrName.Abbreviated+"| appears to be Offline
        LogStep warn,CStr(Now)+tb+|The HCL Domino Server "|+svrName.Abbreviated+"| appears to be Offline
        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 ' if the port is not enforcing Deny Access
        serverList(count)=serverList(count)+"true"
    ElseIf entryNote.HTTP_EnforceAccess(0)="1" Then ' if the port is enforcing Deny Access
        serverList(count)=serverList(count)+"true"
    Else
        If online Then
            If testDB.isopen Then httpSecure=f
            Observation 1002,|The HCL Domino Server "|+svrName.Abbreviated+"| is not properly enforcing Deny Access
            LogStep sec,CStr(Now)+tb+|The HCL Domino Server "|+svrName.Abbreviated+"| is not properly enforcing Deny Access
        End If
    End If
    serverList(count)=serverList(count)+"|"
    If entryNote.LDAP_PortStatus(0)="0" Then ' if the port is not enforcing Deny Access
        serverList(count)=serverList(count)+"true"
    ElseIf entryNote.LDAP_EnforceAccess(0)="1" Then ' if the port is enforcing Deny Access
        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 enforcing Deny Access
            LogStep sec,CStr(Now)+tb+|The HCL Domino Server "|+svrName.Abbreviated+"| is not properly enforcing Deny Access
        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, z 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
        If daNote.DomainEnabled(0)="1" Then
            Observation 1009,|The Directory Assistance configuration "|+daNote.DomainName(0)+|" (|+da
            SetAdminValue "LDAPDA","true"
        End If
    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+"~"+daNote.Getfirstitem("AddressBook_"+CStr(y)).Text
                Else
                    daArray(x)=daNote.Getfirstitem("ServerName_"+CStr(y)).Text+"~"+daNote.Getfirstitem
                End If
                x=x+1
            End If
        Next
        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 splitVar(0)="" Then splitVar(0)=adminServer
    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)
        dbInfo=svrName.Abbreviated+"|" +testDB.Filepath
        LogStep norm,CStr(Now)+tb+|Checking design of Domino Directory - |+testDB.Title+" (" +dbInfo
        If CheckNABDesign(testDB) Then
            If Not IsNumeric(ArrayGetIndex(fullArray,testDB.Title+" (" +dbInfo+"")+"~"+dbInfo)) And No
                ReDim Preserve fullArray(y)
                fullArray(y)=testDB.Title+" (" +dbInfo+"")+"~"+dbInfo
                GetListOfOrgs testDB
                y=y+1
                z=UBound(replIDArray)+1
                ReDim Preserve replIDArray(z)
                replIDArray(z)=testDB.Replicaid
            End If
        Else
            LogStep norm,CStr(Now)+tb+|Invalid Domino Directory design, skipping - |+testDB.Title+"
        End If
    Else
        Observation 1008,|The HCL Domino Directory configured in Directory Assistance cannot be lo
        LogStep warn,CStr(Now)+tb+|The HCL Domino Directory configured in Directory Assistance 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=Erl
        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 currentDate As New NotesDateTime(Now)

```

```

Dim tsDate As NotesDateTime
Dim diff As Long
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
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
    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")
        ' check to see if the user is a current user used Domino server in the past 30 days
        userNote.ActiveUser=""
        If copyItem.Text<>"" Then
            Set tsDate=New NotesDateTime(copyItem.Text)
            diff=currentDate.Timedifference(tsDate)/86400
            If diff<31 Then userNote.ActiveUser="true"
        End If
        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")
        End If
    End While

```

```

        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
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)+|";2);@If(
    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
    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 CCB and CCX licenses
Set userView=db.Getview("counts")
Call userView.Refresh()
userView.Autoupdate=f
'
Dim auCount As Long
Dim dc As NotesDocumentCollection
Set dc=userView.Getalldocumentsbykey("all")
' get the count of ALL Authorized Users
SetAdminValue "ValidUserCount",dc.Count
' get the CCB and CCX counts
Dim keys(1) As String
keys(0)="all"
keys(1)="CCB"
Set dc=userView.Getalldocumentsbykey(keys)
SetAdminValue "ValidUserCountCCB",dc.Count
keys(1)="CCXtrue"
Set dc=userView.Getalldocumentsbykey(keys)
SetAdminValue "ValidUserCountCCXActive",dc.Count
keys(1)="CCXfalse"
Set dc=userView.Getalldocumentsbykey(keys)
SetAdminValue "ValidUserCountCCXInactive",dc.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 E
  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("counts")
Call userView.Refresh()
userView.Autoupdate=f
Set userDC=userView.GetAlldocumentsbykey("all")
If userDC.Count<>0 Then
    Call userDC.Stampall("ServersAccessible",serverGroupList)
    Call userDC.Stampall("ServersAvailable",serverGroupList)
End If
SetAdminValue "ValidUserCount",userDC.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 server manually."
    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" )<>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 your HCL Domino server."
            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),"\" )
    End If
End Sub

```

```

        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 HCL Domino Directory you selected is not accessible. Please
    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 notelItem As NotesItem
Dim serverList As Variant
Dim serverName As NotesName
Dim spitVar 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")
' get the list of Available Servers for cross checking if the Domino Mail server(s) are available
serverList=GetAdminValue("ServerList")
For x=0 To UBound(serverList)
    splitVar=Split(serverList(x),",")
    If splitVar(1)="Available" Then
        serverList(x)=LCase(splitVar(0)) ' convert to lowercase for normalized check later
    Else
        serverList(x)=""
    End If
Next
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
            Set notelItem=nabNote.Getfirstitem("HTTPPassword")
            If Not notelItem Is Nothing Then If notelItem.Text<>"" Then pwAvail=t
        End If
    End If

```

```

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
End If
If Not userNote.Hasitem("UserType") And nabNote.Hasitem("HTTPPassword") Then
    If nabNote.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
    Call userNote.Removeitem("MailUser")
    Call userNote.Removeitem("MailServer")
    Call userNote.Removeitem("MailFile")
    If nabNote.MailSystem(0)="1" And nabNote.MailServer(0)<>"" Then
        ' check if the mail server is available
        Set serverName=New NotesName(nabNote.MailServer(0))
        If IsNumeric(ArrayGetIndex(serverList,LCASE(serverName.Abbreviated))) Then
            userNote.MailUser="true"
            userNote.NABNoteMailServer=nabNote.MailServer(0)
            userNote.NABNoteMailFile=nabNote.MailFile(0)
        End If
    End If
    userNote.FullName=nabNote.FullName(0)
    userNote.InternetAddress=nabNote.InternetAddress(0)
    userNote.HTTPPasswordSet="false"
    If nabNote.HasItem("HTTPPassword") Then
        Set noteltem=nabNote.Getfirstitem("HTTPPassword")
    End If
End If

```

```

        If Not notelItem Is Nothing Then If notelItem.Text<>"" Then userNote.HTTPPasswordSet="t
    End If
    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.ExternalORG
        Next
    End If
    If deepLog Then
        count=count+1
        logTXT=CStr(count)+" of "+CStr(navCount)+" - Person = "+userNote.FullName(0)+" (" +user
        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 a
End If
Exit Function
End Function

```

## Name:

Last Modification:

## 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=t
Const deepLog=f

```

## slUtilities

07/04/2023 10:52:17 AM

```

Const norm=0
Const info=1
Const sec=2
Const warn=3
Function GetAdminValue(fldName As String) As Variant
    Dim empty(0) As String
    fldName=FullTrim(fldName)
    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
        If admin.Getfirstitem(fldName).text<>"" Then
            GetAdminValue=admin.Getfirstitem(fldName).Values
        End If
    End If
End Function
Sub SetAdminValue (fldName As String,value As Variant)
    Dim item As NotesItem
    Dim summaryFields(8) As String
    fldName=FullTrim(fldName)
    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 ValidateHCLID(id As String) As Boolean

%REM

This function will attempt to validate the structure of the HCL Account ID.

Most HCL IDs have the following structure:

HCL123456

- All HCL IDs start with "HCL"

- Most if not all HCL IDs have a numerical component with 5-10 digits

- There are some odd occurrences where there is also a dash ( - ) as part of the HCL ID (e.g. HCL12345-2)

We need to see if what the user entered follows most if not all of these HCL ID styles. If it does not, we need to give the user a chance to correct what was entered. If the user states that the HCL ID is correct, use what they entered in the function.

%END REM

cr=Chr(10)

tb=Chr(9)

ValidateHCLID=f

Dim tmpString As String

Dim msg As String

tmpString=LCase(id)

If Left(tmpString,3)<>"hcl" Then

msg=msg+cr+| - HCL Account #s start with "HCL"|

Else

tmpString=Right(tmpString, Len(tmpString)-3)

End If

If Len(tmpString)<5 Then msg=msg+cr+| - The length of the HCL Account #s has a minimum length of 8 characters

If Not IsNumeric(tmpString) Then msg=msg+cr+| - The HCL Account # is typically a numerical value (e.g. 123456)

If msg<>"" Then

If MsgBox("The HCL Account Number that you provided does not conform the standard issued values",vbYesNo,"Question")=vbYes Then

cr+cr+msg+cr+cr+"Do you want to continue using the value you provided",36,"Question")=6 Then ValidateHCLID=t

Else

ValidateHCLID=t

End If

End Function

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

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 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:

### **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

```

### **(MakeHistory)**

MakeHistory  
 07/04/2023 10:52:16 AM  
 [Not Assigned]  
 Yes  
 LotusScript  
 Enabled  
 Manually From Actions Menu  
 None

```

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

07/04/2023 10:52:16 AM

[Not Assigned]

Yes

LotusScript

Enabled

Manually From Actions Menu

None

**LotusScript Code:**

```

Option Public
Option Declare
Use "slCore"
Sub Initialize
    Dim askQuestions As Boolean
    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:

**(PullInfoDAGroups)**

PullInfoDAGroups

07/04/2023 10:52:16 AM

[Not Assigned]

Yes

Type:	LotusScript
State:	Enabled
Trigger:	Manually From Actions Menu
Acts On:	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:	(PullInfoEntitlement)
Last Modification:	PullInfoEntitlement
Comment:	07/04/2023 10:52:16 AM
Shared Agent:	[Not Assigned]
Type:	Yes
State:	LotusScript
Trigger:	Enabled
Acts On:	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

07/04/2023 10:52:16 AM

[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 routine"
    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 additional database"
                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 = False
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 information"
                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

```

**Name:**

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

**ViewReportFM**

07/04/2023 10:52:15 AM  
 [Not Assigned]  
 Yes  
 Formula  
 Enabled  
 Manually From Agent List  
 None  
 check :=  
 @Environment("GetInfoCanceled");@If(check = ""; @Success;  
 @Return(@False));  
 @SetTargetFrame("Main");  
 @Command([OpenFrameset];  
 "fsReport");

**Name:**

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

**ViewReportLS**

07/04/2023 10:52:16 AM  
 [Not Assigned]  
 Yes  
 LotusScript  
 Enabled  
 Manually From Agent List  
 None

**LotusScript Code:**

```
Option Public
Option Declare
Use "slUtilities"
Sub Initialize
  cr=Chr(10)
  tb=Chr(9)
  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)
  Dim pdCount As String
  Dim daCount As String
  Dim vuCount As String
  pdCount=session.Getenvironmentstring("PersonDocCount")
  daCount=session.Getenvironmentstring("DAUserCount")
  vuCount=session.Getenvironmentstring("ValidUserCount")
  Call session.Setenvironmentvar("GetInfoCanceled", "")
  Call session.Setenvironmentvar("PersonDocCount", "")
  Call session.Setenvironmentvar("DAUserCount", "")
  Call session.Setenvironmentvar("ValidUserCount", "")
  If GetAdminValue ("CompanyName")(0)="" Then
    If Not ws.dialogbox("dlgUserInfo",t,t,f,f,f,f,"User Information",dialog,t,f,t) Then
      Call session.Setenvironmentvar("GetInfoCanceled", "true")
    Exit Sub
  End If
  If dialog.CompanyName(0)="" Then msgStr=msgStr+cr+tb+"- Company Name"
  If dialog.HCLAccount(0)="" Then msgStr=msgStr+cr+tb+"- HCL Account #"
  If msgStr<>"" Then
    MsgBox "Not all information was provided. All information is required."+cr+msgStr
```

```

        Call session.Setenvironmentvar("GetInfoCanceled", "true")
        Exit Sub
    End If
    If Not ValidateHCLID(dialog.HCLAccount(0)) Then Call session.Setenvironmentvar("GetInfoCanceled"
    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",CLng(pdCount)
SetAdminValue "DAUserCount",CLng(daCount)
SetAdminValue "ValidUserCount",CLng(vuCount)
errHandle:
    If Err<>0 Then MsgBox("Error in function ' " & GetThreadInfo(1) & ": " & GetThreadInfo(1) & "' at " & Erl & "
    Exit Sub
End Sub

```

## Page Information

### **Name:**

Last Modification:

Comment:

Treat contents as HTML:

### **Subcomponents:**

[None]

### **JavaScript & HTML Code:**

[None]

### **LotusScript Code:**

[None]

### **pgBlank**

07/04/2023 10:52:14 AM

[Not Assigned]

No

### **Name:**

Alias:

Last Modification:

Comment:

Treat contents as HTML:

### **Subcomponents:**

[None]

### **JavaScript & HTML Code:**

[None]

### **LotusScript Code:**

[None]

### **pgFooter**

black

07/04/2023 10:52:16 AM

[Not Assigned]

No

### **Name:**

Last Modification:

Comment:

Window Title Formula:

Treat contents as HTML:

### **Subcomponents:**

[None]

### **pgHeader**

07/04/2023 10:52:16 AM

[Not Assigned]

@DbTitle

No

## 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
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:
    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 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**

07/04/2023 10:52:14 AM

[Not Assigned]

No

**Name:**

Last Modification:

Comment:

Treat contents as HTML:

**Subcomponents:****Hotspot**

Type:

Caption:

Is Scripted:

**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

If valCheck.Organization&lt;&gt;currentUser.Organization Then Exit Sub

Msgbox "The design of the Domino License Analysis Utility has been re-signed."+Chr(10)+Chr(10)+"the uti

Call ws.CurrentDocument.Close

**pgNotSigned**

07/04/2023 10:52:16 AM

[Not Assigned]

No

Action Hotspot

Re-sign Design

Yes, see below

## Frameset Information

**Name:**

Last Modification:

Comment:

Available to public access users:

Window Title Formula:

**HTML Code:**

&lt;frameset border="0" rows="" cols="150,80%"&gt;

**fslInner**

07/04/2023 10:52:14 AM

[Not Assigned]

No

@DbTitle

```
<frame name="Sidebar" frameborder="0" scrolling="no" noresize>
<frame name="Main" scrolling="auto">
</frameset>
```

**Name:**

Last Modification:

Comment:

Available to public access users:

Window Title Formula:

**HTML Code:**

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

**fsLaunch**

07/04/2023 10:52:14 AM

[Not Assigned]

No

[Not Assigned]

**Name:**

Last Modification:

Comment:

Available to public access users:

Window Title Formula:

**HTML Code:**

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

**fsLog**

07/04/2023 10:52:14 AM

[Not Assigned]

No

[Not Assigned]

**Name:**

Last Modification:

Comment:

Available to public access users:

Window Title Formula:

**HTML Code:**

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

**fsReport**

07/04/2023 10:52:14 AM

[Not Assigned]

No

[Not Assigned]

**Name:**

Last Modification:

Comment:

Available to public access users:

Window Title Formula:

**HTML Code:**

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

**fsResults**

07/04/2023 10:52:14 AM

[Not Assigned]

No

[Not Assigned]

**Name:**

Last Modification:

Comment:

Available to public access users:

Window Title Formula:

**fsSidebar**

07/04/2023 10:52:14 AM

[Not Assigned]

No

[Not Assigned]

**HTML Code:**

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

**Name:**

Last Modification:

Comment:

Available to public access users:

Window Title Formula:

**fsSidebar2**

07/04/2023 10:52:14 AM

[Not Assigned]

No

[Not Assigned]

**HTML Code:**

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

**Name:**

Last Modification:

Comment:

Available to public access users:

Window Title Formula:

**fsSidebar3**

07/04/2023 10:52:14 AM

[Not Assigned]

No

[Not Assigned]

**HTML Code:**

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

**Name:**

Last Modification:

Comment:

Available to public access users:

Window Title Formula:

**fsTools**

07/04/2023 10:52:14 AM

[Not Assigned]

No

[Not Assigned]

**HTML Code:**

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

**Name:**

Last Modification:

Comment:

Available to public access users:

Window Title Formula:

**Launch**

07/04/2023 10:52:15 AM

[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.gif**

07/04/2023 10:52:18 AM  
[Not Assigned]  
1  
1  
No

**Name:**

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

**actionbarblack.png**

07/04/2023 10:52:15 AM  
[Not Assigned]  
1  
1  
No

**Name:**

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

**actionbarwhite.gif**

07/04/2023 10:52:15 AM  
[Not Assigned]  
1  
1  
No

**Name:**

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

**actionbarwhite.png**

07/04/2023 10:52:15 AM  
[Not Assigned]  
1  
1  
No

**Name:**

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

**blue-bar.gif**

07/04/2023 10:52:17 AM  
[Not Assigned]  
1  
1  
No

**Name:**

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

**blue-bar.png**

07/04/2023 10:52:15 AM  
[Not Assigned]  
1  
1  
No

**Name:**

Last Modification:  
Comment:  
Images Across:  
Images Down:

**buttonbg.gif**

07/04/2023 10:52:18 AM  
[Not Assigned]  
1  
1

Colorize Grays:

No

**Name:**

Last Modification:

Comment:

Images Across:

Images Down:

Colorize Grays:

**buttonbg.png**

07/04/2023 10:52:16 AM

[Not Assigned]

1

1

No

**Name:**

Last Modification:

Comment:

Images Across:

Images Down:

Colorize Grays:

**circle-dot.gif**

07/04/2023 10:52:18 AM

[Not Assigned]

1

1

No

**Name:**

Last Modification:

Comment:

Images Across:

Images Down:

Colorize Grays:

**circle-dot.png**

07/04/2023 10:52:16 AM

[Not Assigned]

1

1

No

**Name:**

Last Modification:

Comment:

Images Across:

Images Down:

Colorize Grays:

**close.gif**

07/04/2023 10:52:18 AM

[Not Assigned]

1

1

No

**Name:**

Last Modification:

Comment:

Images Across:

Images Down:

Colorize Grays:

**close.png**

07/04/2023 10:52:15 AM

[Not Assigned]

1

1

No

**Name:**

Last Modification:

Comment:

Images Across:

Images Down:

Colorize Grays:

**denyaccess.gif**

07/04/2023 10:52:18 AM

[Not Assigned]

1

1

No

**Name:**

Last Modification:

Comment:

**denyaccess.png**

07/04/2023 10:52:15 AM

[Not Assigned]

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

<b>Name:</b>	<b>edit.gif</b>
Last Modification:	07/04/2023 10:52:18 AM
Comment:	[Not Assigned]
Images Across:	1
Images Down:	1
Colorize Grays:	No

<b>Name:</b>	<b>edit.png</b>
Last Modification:	07/04/2023 10:52:16 AM
Comment:	[Not Assigned]
Images Across:	1
Images Down:	1
Colorize Grays:	No

<b>Name:</b>	<b>export.gif</b>
Last Modification:	07/04/2023 10:52:19 AM
Comment:	[Not Assigned]
Images Across:	1
Images Down:	1
Colorize Grays:	No

<b>Name:</b>	<b>export.png</b>
Last Modification:	07/04/2023 10:52:15 AM
Comment:	[Not Assigned]
Images Across:	1
Images Down:	1
Colorize Grays:	No

<b>Name:</b>	<b>hcl-software-logo.gif</b>
Last Modification:	07/04/2023 10:52:19 AM
Comment:	[Not Assigned]
Images Across:	1
Images Down:	1
Colorize Grays:	No

<b>Name:</b>	<b>hcl-software-logo.png</b>
Last Modification:	07/04/2023 10:52:16 AM
Comment:	[Not Assigned]
Images Across:	1
Images Down:	1
Colorize Grays:	No

<b>Name:</b>	<b>history.gif</b>
--------------	--------------------

Last Modification:	07/04/2023 10:52:19 AM
Comment:	[Not Assigned]
Images Across:	1
Images Down:	1
Colorize Grays:	No

<b>Name:</b>	<b>history.png</b>
Last Modification:	07/04/2023 10:52:15 AM
Comment:	[Not Assigned]
Images Across:	1
Images Down:	1
Colorize Grays:	No

<b>Name:</b>	<b>home.gif</b>
Last Modification:	07/04/2023 10:52:17 AM
Comment:	[Not Assigned]
Images Across:	1
Images Down:	1
Colorize Grays:	No

<b>Name:</b>	<b>home.png</b>
Last Modification:	07/04/2023 10:52:15 AM
Comment:	[Not Assigned]
Images Across:	1
Images Down:	1
Colorize Grays:	No

<b>Name:</b>	<b>info.gif</b>
Last Modification:	07/04/2023 10:52:19 AM
Comment:	[Not Assigned]
Images Across:	1
Images Down:	1
Colorize Grays:	No

<b>Name:</b>	<b>info.png</b>
Last Modification:	07/04/2023 10:52:15 AM
Comment:	[Not Assigned]
Images Across:	1
Images Down:	1
Colorize Grays:	No

<b>Name:</b>	<b>loading.gif</b>
Last Modification:	07/04/2023 10:52:17 AM
Comment:	[Not Assigned]
Images Across:	1
Images Down:	1
Colorize Grays:	No

**Name:**

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

**logo\_header.gif**

07/04/2023 10:52:18 AM  
[Not Assigned]  
1  
1  
No

**Name:**

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

**logo\_header.png**

07/04/2023 10:52:16 AM  
[Not Assigned]  
1  
1  
No

**Name:**

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

**logo\_tiny.gif**

07/04/2023 10:52:19 AM  
[Not Assigned]  
1  
1  
No

**Name:**

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

**logo\_tiny.png**

07/04/2023 10:52:16 AM  
[Not Assigned]  
1  
1  
No

**Name:**

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

**log.gif**

07/04/2023 10:52:19 AM  
[Not Assigned]  
1  
1  
No

**Name:**

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

**log.png**

07/04/2023 10:52:15 AM  
[Not Assigned]  
1  
1  
No

**Name:**

Last Modification:  
Comment:  
Images Across:  
Images Down:

**observation.gif**

07/04/2023 10:52:19 AM  
[Not Assigned]  
1  
1



Colorize Grays:

No

**Name:**

Last Modification:

Comment:

Images Across:

Images Down:

Colorize Grays:

**observation.png**

07/04/2023 10:52:16 AM

[Not Assigned]

1

1

No

**Name:**

Last Modification:

Comment:

Images Across:

Images Down:

Colorize Grays:

**print.gif**

07/04/2023 10:52:19 AM

[Not Assigned]

1

1

No

**Name:**

Last Modification:

Comment:

Images Across:

Images Down:

Colorize Grays:

**print.png**

07/04/2023 10:52:15 AM

[Not Assigned]

1

1

No

**Name:**

Last Modification:

Comment:

Images Across:

Images Down:

Colorize Grays:

**refresh.gif**

07/04/2023 10:52:19 AM

[Not Assigned]

1

1

No

**Name:**

Last Modification:

Comment:

Images Across:

Images Down:

Colorize Grays:

**refresh.png**

07/04/2023 10:52:15 AM

[Not Assigned]

1

1

No

**Name:**

Last Modification:

Comment:

Images Across:

Images Down:

Colorize Grays:

**save.gif**

07/04/2023 10:52:19 AM

[Not Assigned]

1

1

No

**Name:**

Last Modification:

Comment:

**save.png**

07/04/2023 10:52:15 AM

[Not Assigned]

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

<b>Name:</b>	<b>select.gif</b>
Last Modification:	07/04/2023 10:52:19 AM
Comment:	[Not Assigned]
Images Across:	1
Images Down:	1
Colorize Grays:	No

<b>Name:</b>	<b>select.png</b>
Last Modification:	07/04/2023 10:52:15 AM
Comment:	[Not Assigned]
Images Across:	1
Images Down:	1
Colorize Grays:	No

<b>Name:</b>	<b>twisty.gif</b>
Last Modification:	07/04/2023 10:52:16 AM
Comment:	[Not Assigned]
Images Across:	2
Images Down:	1
Colorize Grays:	No

<b>Name:</b>	<b>users.gif</b>
Last Modification:	07/04/2023 10:52:19 AM
Comment:	[Not Assigned]
Images Across:	1
Images Down:	1
Colorize Grays:	No

<b>Name:</b>	<b>users.png</b>
Last Modification:	07/04/2023 10:52:15 AM
Comment:	[Not Assigned]
Images Across:	1
Images Down:	1
Colorize Grays:	No

<b>Name:</b>	<b>view.gif</b>
Last Modification:	07/04/2023 10:52:19 AM
Comment:	[Not Assigned]
Images Across:	1
Images Down:	1
Colorize Grays:	No

<b>Name:</b>	<b>view.png</b>
--------------	-----------------

Last Modification:	07/04/2023 10:52:15 AM
Comment:	[Not Assigned]
Images Across:	1
Images Down:	1
Colorize Grays:	No

<b>Name:</b>	<b>warning.gif</b>
Last Modification:	07/04/2023 10:52:19 AM
Comment:	[Not Assigned]
Images Across:	1
Images Down:	1
Colorize Grays:	No

<b>Name:</b>	<b>warning.png</b>
Last Modification:	07/04/2023 10:52:15 AM
Comment:	[Not Assigned]
Images Across:	1
Images Down:	1
Colorize Grays:	No

<b>Name:</b>	<b>wizard.gif</b>
Last Modification:	07/04/2023 10:52:19 AM
Comment:	[Not Assigned]
Images Across:	1
Images Down:	1
Colorize Grays:	No

<b>Name:</b>	<b>wizard.png</b>
Last Modification:	07/04/2023 10:52:15 AM
Comment:	[Not Assigned]
Images Across:	1
Images Down:	1
Colorize Grays:	No

<b>Name:</b>	<b>\$DBIcon</b>
Last Modification:	07/04/2023 10:52:19 AM
Comment:	[Not Assigned]
Images Across:	1
Images Down:	1
Colorize Grays:	No

## Other Information

<b>Type:</b>	<b>Icon</b>
Last Modification:	07/04/2023 10:52:12 AM
<b>Type:</b>	<b>"Using Application"</b>
	<b>Document</b>
Last Modification:	07/04/2023 10:52:14 AM

**JavaScript & HTML Code:**

[None]

**LotusScript Code:**

[None]

**Type:**

Last Modification:

**JavaScript & HTML Code:**

[None]

**LotusScript Code:**

[None]

**Type:**

Last Modification:

**LotusScript Code:**

[None]

**"About Application"  
Document**

07/04/2023 10:52:17 AM

**Database Script**

07/04/2023 10:52:16 AM