

Re: active directory question

Source:

<http://www.tech-archive.net/Archive/Scripting/microsoft.public.scripting.vbscript/2008-05/msg00405.html>

- *From:* "Richard Mueller [MVP]" <rlmueller-nospam@xxxxxxxxxxxxxxxxxxxxxxxxxxxxx>
 - *Date:* Fri, 16 May 2008 18:07:38 -0500
-

The first recommendation has to be to not use "On Error Resume Next" throughout. I know you trap many possible errors early in the script, but a lot can go wrong later (and perhaps does). My approach is to only use "On Error Resume Next" for the statements I expect might raise an error, then handle the error (as you do in several places), then restore normal error handling with "On Error GoTo 0". This makes troubleshooting much easier.

You seem to place the first trustee, in the form <domain>\<nt name>, in row 3, column 1. Later when you process trustees you start with row 2, which is the header "Login\Group Name". Seems like you should start with x=3.

After saving the trustee in the form <domain>\<nt name>, you later parse this and replace it with just <nt name> (I think). I would not do that.

Later you seem to use ADO to find the trustee. First, the statements that setup the ADO objects should be outside the loop, as the binding to the objects only needs to be done once. The statements:

```
Set objCommand = CreateObject("ADODB.Command")
Set objConnection = CreateObject("ADODB.Connection")
objConnection.Provider = "ADsDSOObject"
objConnection.Open "Active Directory Provider"
objCommand.ActiveConnection = objConnection
strBase = "<LDAP://dc=test,dc=test1,dc=com>"
strAttributes = "sAMAccountName,cn,member,objectClass"
objCommand.CommandText = strQuery
objCommand.Properties("Page Size") = 100
objCommand.Properties("Timeout") = 30
objCommand.Properties("Cache Results") = False
```

can all be executed once, outside the loop. Next, the trustee name retrieved from the ACE is technically not the Common Name, but rather the NT name (generally the value of the sAMAccountName attribute). In most cases the values of the cn and sAMAccountName attributes are the same for groups, but this is not required.

Next, you retrieve the value of the objectCategory attribute from the recordset, but it should not be there, as this was not included in the comma

Re: active directory question

delimited list of attributes values to be retrieved. Then you compare the value of the objectClass attribute to the string "Top;group". This should never be true, as the objectClass attribute is multi-valued and ADO retrieves it as an array. One element of the array will be "group". However, testing for this should not be necessary as your filter already restricts the recordset to objects where "(objectCategory=group)".

I'm not sure what happens later, as you seem to add worksheets for each group and user. I cannot follow.

Rather than using ADO to search for the trustee it could make sense to use the NameTranslate object to convert the original trustee name in the form <domain>\<nt name> into the Distinguished Name of the trustee. However, you must then bind to the object, check if it is a group, then enumerate the members. Maybe using ADO is more efficient in this case, as you can retrieve the value of the member attribute without the need to bind to each trustee. However, I think the filter should be:

```
strFilter = "(&(objectCategory=group)(sAMAccountName=" &  
objExcel.Worksheets(1).Cells(x, 1).Value & ")")"
```

The only attribute you need retrieve is "member". If the trustee is not a group, the recordset will be empty.

Perhaps the point of your post is that the member attribute is a collection of DN's, the Distinguished Names of the members of the group. It sounds like you want the value of the sAMAccountName of each member. This requires that you bind to each member. You would do this with the DN.

This makes me think the best approach (if I understand what you are doing) is to use ADO to search for all objects that are members of the trustee (assuming the trustee is a group). For this I would use NameTranslate to convert the trustee name in the form <domain>\<nt name> into the Distinguished Name (DN). Then I would use ADO to find all objects where the memberOf attribute is equal to the DN. Now ADO can retrieve any attributes of the member object desired, such as sAMAccountName (what you refer to as the logon name).

In brief, eliminate the code that removes <domain> from the trustee name, as that is needed for NameTranslate. For information on using NameTranslate see this link:

<http://www.rlmueller.net/NameTranslateFAQ.htm>

At the very least, add "On Error GoTo 0" after you have trapped all expected errors. Better yet, use "On Error Resume Next" before the statements that might raise errors, then test for the errors (as you do), then restore normal error handling with "On Error GoTo 0".

Once you have converted <domain>\<nt name> into a DN, and perhaps assigned the value to the variable strDN, the ADO filter could be:

Re: active directory question

Re: active directory question

```
strFilter = "(memberOf=" & strDN & ")"
```

If the trustee is not a group, the recordset will be empty, as no objects will satisfy the filter clause. The only attribute you need to retrieve is the sAMAccountName, which will be the logon name of the member of the trustee (if the trustee is a group).

I hope this helps.

--
Richard Mueller
MVP Directory Services
Hilltop Lab – <http://www.rlmueller.net>
--

"JayJ" <JayJ@xxxxxxxxxxxxxxxxxxxxxxxxxxxx> wrote in message
<news:AB71D124-0FA3-49C7-AC5B-815452AC87CD@xxxxxxxxxxxxxxxxxxxx>

With the help of this newsgroup I have a script that will pull the the groups off a specified folder output them and the members of each group to separate sheets in an excel spreadsheet. I also want to output the logon names of each member. To do this I have to access the User properties instead of the Group.

The format of the name in the spreadsheet is smith,john – which is the display name in active directory. Can i reference this name from the spreadsheet and query AD to output the logon name of each user? This location is referenced in the script by this –
objExcel.WorkSheets(w).cells(y,1).value

I don't know how to put this back into an AD query to pull logon names for the user display name that is in that location – also has to loop through because i don't know how many users will be in each group.

Any help is appreciated – script is copied below.

```
Dim objCommand, objConnection, strBase, strFilter, strAttributes  
Dim strQuery, objRecordset, strName, strCN  
Dim excelgroups, objExcel, objWshNet, strFoldername, UNCPathName,  
DrvLetter,  
strComputerName  
Set objExcel = CreateObject("Excel.Application")  
On Error resume Next  
  
objExcel.Visible = True  
objExcel.Workbooks.Add
```

Re: active directory question

```
objExcel.Cells(2, 1).Value = "Login\Group Name"  
objExcel.Cells(2, 1).Font.Bold = True  
objExcel.Cells(2, 2).Value = "Access Allowed\Denied"  
objExcel.Cells(2, 2).Font.Bold = TRUE  
objExcel.Cells(2, 3).Value = "Permission Assigned"  
objExcel.Cells(2, 3).Font.Bold = TRUE  
objExcel.WorkSheets(1).name = "Permissions List"
```

```
UNCPathName = InputBox("please supply the UNC path to the shared folder")  
DrvLetter = InputBox("Please supply unused driver letter followed by a  
colon")
```

```
set objWshNet = WScript.CreateObject("Wscript.Network")  
objWshNet.MapNetworkDrive DrvLetter, UNCPathName
```

```
If Err.Number <> 0 Then  
Wscript.Echo "Error: " & Err.Number & vbCrLf &  
Err.Description & " 0"  
End If
```

```
If Err.Number <> 0 Then  
Wscript.Echo "Error: " & Err.Number & vbCrLf &  
Err.Description & " 1"  
End If  
objExcel.Cells(1, 1).Value = UNCPathName  
objExcel.Cells(1, 1).Font.Bold = TRUE  
SE_DACL_PRESENT = &h4  
ACCESS_ALLOWED_ACE_TYPE = &h0  
ACCESS_DENIED_ACE_TYPE = &h1  
If Err.Number <> 0 Then  
Wscript.Echo "Error: " & Err.Number & vbCrLf &  
Err.Description & " 2"  
End If
```

```
Set objWMIService = GetObject("winmgmts:")
```

```
If Err.Number <> 0 Then  
Wscript.Echo "Error: " & Err.Number & vbCrLf &  
Err.Description & " 3"  
End If
```

```
Set objFolderSecuritySettings = _  
objWMIService.Get("Win32_LogicalFileSecuritySetting.path=" & DrvLetter &  
"\")
```

```
If Err.Number <> 0 Then  
Wscript.Echo "Error: " & Err.Number & vbCrLf &  
Err.Description & " 4"  
End If
```

Re: active directory question

```
intRetVal = objFolderSecuritySettings.GetSecurityDescriptor(objSD)

If Err.Number <> 0 Then
Wscript.Echo "Error: " & Err.Number & vbCrLf & _
Err.Description & " 5"
End If

intControlFlags = objSD.ControlFlags

If intControlFlags AND SE_DACL_PRESENT Then

arrACEs = objSD.DACL
X=3
For Each objACE in arrACEs

objExcel.Cells(x, 1).Value = _
objACE.Trustee.Domain & "\" & objACE.Trustee.Name
If objACE.AceType = ACCESS_ALLOWED_ACE_TYPE Then
objExcel.Cells(x, 2).Value = _
vbTab & "Allowed:"
ElseIf objACE.AceType = ACCESS_DENIED_ACE_TYPE Then
objExcel.Cells(x, 2).Value = _
vbTab & "Denied:"
End If
If objACE.AccessMask = "1245631" Then
objExcel.Cells(x, 3).Value = "Modify"
End If
If objACE.AccessMask = "1179785" Then
objExcel.Cells(x, 3).Value = "Read Only"
End If
If objACE.AccessMask = "1179817" Then
objExcel.Cells(x, 3).Value = "Read & Execute"
End If
If objACE.AccessMask = "2032127" Then
objExcel.Cells(x, 3).Value = "Full Control"
End If

X=X+1

Next
Else
WScript.Echo "No DACL present in security descriptor"
End If

Set objRange = objExcel.Range("A1")
objRange.Activate

Set objRange = objExcel.ActiveCell.EntireColumn
objRange.Autofit()
```

Re: active directory question

```
Set objRange = objExcel.Range("B1")
objRange.Activate
Set objRange = objExcel.ActiveCell.EntireColumn
objRange.Autofit()
```

```
Set objRange = objExcel.Range("A1").SpecialCells(11)
Set objRange2 = objExcel.Range("C1")
Set objRange3 = objExcel.Range("A1")
```

```
x=2
Do Until objExcel.Cells(x,1).Value = ""
arrSecCon= Split(objExcel.Cells(x,1).Value, "\")
CellValue=objExcel.Cells(x,1).Value
objExcel.Cells(x,1).Value=CellValue
x=x+1
loop
```

```
w=2
x=2
```

```
Do Until objExcel.Worksheets(1).Cells(x,1).Value = ""
```

```
Set objCommand = CreateObject("ADODB.Command")
Set objConnection = CreateObject("ADODB.Connection")
objConnection.Provider = "ADsDSOObject"
objConnection.Open "Active Directory Provider"
objCommand.ActiveConnection = objConnection
strBase = "<LDAP://dc=test.dc=test1.dc=com>"
strFilter = "(&(objectCategory=group)(cn=" &
objExcel.Worksheets(1).Cells(x,1).Value & ")")"
strAttributes = "sAMAccountName,cn,member,objectClass"
strQuery = strBase & ";" & strFilter & ";" & strAttributes
&
";subtree"
objCommand.CommandText = strQuery
objCommand.Properties("Page Size") = 100
objCommand.Properties("Timeout") = 30
objCommand.Properties("Cache Results") = False
Set objRecordSet = objCommand.Execute
objExcel.Worksheets(1).cells(x,
2).value=objRecordSet.Fields("objectCategory").Value
If objRecordSet.Fields("objectClass").Value = "Top:group" Then
Do Until objRecordSet.EOF
```

```
MbrName = objRecordSet.Fields("sAMAccountName").Value
Wscript.echo "Beginning of enumeration of group " & MbrName
y=2
```

Re: active directory question

```
arrUsers = objRecordSet.Fields("member").Value

If IsNull(arrUsers) Then
Wscript.Echo "-- No users assigned to group"
Else

If w>=4 Then

objExcel.worksheets.Add

objExcel.WorkSheets(w).move objExcel.WorkSheets(w-1)

End If
objExcel.WorkSheets(w).Activate
objExcel.Cells(1, 1).Font.Bold = TRUE
objExcel.WorkSheets(w).Cells(1, 1).Value = MbrName

Set objRange = objExcel.Range("A1")
objRange.Activate
Set objRange = objExcel.ActiveCell.EntireColumn
objRange.Autofit()
Set objRange = objExcel.Range("B1")
objRange.Activate
Set objRange = objExcel.ActiveCell.EntireColumn
objRange.Autofit()

For Each strUser In arrUsers

'Get the position of "OU="
OU_pos = Instr(strUser, "OU=")
strUser1 = strUser

If objRecordSet.Fields("objectClass").Value = "Top;group" Then

'objExcel.WorkSheets(w).cells(y,1).value=strUser

'Lets get only the info we are interested in
'Starting at position 4 and up to the location of "OU"

strUser1 = Replace(strUser1, "\", "")
objExcel.WorkSheets(w).cells(y,1).value=Mid(strUser1, 4,
OU_pos - 5)

y=y+1
End If
Next

End If
objRecordSet.MoveNext
```

Re: active directory question

```
objExcel.WorkSheets(w).name = MbrName
```

```
w=W+1
```

```
Loop
```

```
End IF
```

```
x=x+1
```

```
loop
```

```
objWshNet.removeNetworkDrive DrvLetter, True, True
```

```
objConnection.Close
```

```
wscript.quit
```