

Re: LDAP query information

Source:

<http://www.tech-archive.net/Archive/Windows/microsoft.public.windows.server.scripting/2006-07/msg00081.html>

- *From:* Vishn <Vishn@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx>
 - *Date:* Tue, 11 Jul 2006 09:32:02 -0700
-

It gave me the below error message when I ran the script. I suppose its some issue with the Type declaration.

Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

O:\>c:

C:\>cd compactcs

C:\compactcs>cscript compactcs.vbs

C:\compactcs>cscript //nologo compactcs.vbs >report.txt
C:\compactcs\compactcs.vbs(57, 5) Microsoft VBScript runtime error: Type mismatch
h

C:\compactcs>

This is how I had saved the script
Option Explicit

```
Dim objRootDSE, strDNSDomain, adoCommand, adoConnection  
Dim strBase, strFilter, strAttributes, strQuery, adoRecordset  
Dim strNTName, lngUSN, arrCanonical, strCanonical  
Dim strDN, strDesc, strRole, strManaged, strOS, strOSVer  
Dim strSP, strLocation, strCreated, strChanged
```

```
' Determine DNS domain name.  
Set objRootDSE = GetObject("LDAP://RootDSE")  
strDNSDomain = objRootDSE.Get("defaultNamingContext")
```

```
' Use ADO to search Active Directory.  
Set adoCommand = CreateObject("ADODB.Command")  
Set adoConnection = CreateObject("ADODB.Connection")  
adoConnection.Provider = "ADsDSOObject"  
adoConnection.Open "Active Directory Provider"
```

Re: LDAP query information

adoCommand.ActiveConnection = adoConnection

' Search entire domain.

strBase = "<LDAP:// & strDNSDomain & ">"

' Filter on computer object.

strFilter = "(objectCategory=computer)"

' Comma delimited list of attributes.

strAttributes = "sAMAccountName.canonicalName.distinguishedName."
& "description.machineRole.managedBy.operatingSystem."
& "location.operatingSystemVersion.operatingSystemServicePack."
& "whenCreated.whenChanged"

' Construct LDAP syntax query.

strQuery = strBase & ":" & strFilter & ":" & strAttributes & ":'subtree"

' Execute the query.

adoCommand.CommandText = strQuery

adoCommand.Properties("Page Size") = 100

adoCommand.Properties("Timeout") = 30

adoCommand.Properties("Cache Results") = False

Set adoRecordset = adoCommand.Execute

' Enumerate the recordset and display values.

Do Until adoRecordset.EOF

strNTName = adoRecordset.Fields("sAMAccountName").Value

arrCanonical = adoRecordset.Fields("canonicalName").Value

strCanonical = arrCanonical(0)

strDN = adoRecordset.Fields("distinguishedName").Value

strDesc = adoRecordset.Fields("description").Value

strRole = adoRecordset.Fields("machineRole").Value

strManaged = adoRecordset.Fields("managedBy").Value

strLocation = adoRecordset.Fields("location").Value

strOS = adoRecordset.Fields("operatingSystem").Value

strOSVer = adoRecordset.Fields("operatingSystemVersion").Value

strSP = adoRecordset.Fields("operatingSystemServicePack").Value

strCreated = adoRecordset.Fields("whenCreated").Value

strChanged = adoRecordset.Fields("whenChanged").Value

Wscript.Echo strNTName & ":' & strCanonical & ":' & strDN

& ":' & strDesc & ":' & strRole & ":' & strManaged

& ":' & strLocation & ":' & strOS & ":' & strOSVer

& ":' & strSP & ":' & strCreated & ":' & strChanged

adoRecordset.MoveNext

Loop

adoRecordset.Close

adoConnection.Close

' Clean up.

Set objRootDSE = Nothing

Set adoCommand = Nothing

Re: LDAP query information

Set adoConnection = Nothing
Set adoRecordset = Nothing

Thanks for your help!

"Richard Mueller" wrote:

Hi.

Yes, the script uses ADO to query AD directly, which is very efficient – no need to bind to each computer object. If the program is saved in a file called Computers.vbs, run it at a command prompt with the following command to redirect the output to a text file:

cscript //nologo Computers.vbs > report.txt

If Computers.vbs is not in the current directory, include the path to the file. The "/nologo" parameter suppresses WSH info. You can add or remove attributes, change the order the values are output, use a different delimiter, whatever.

==

Richard
Microsoft MVP Scripting and ADSI
Hilltop Lab – <http://www.rlmuellet.net>

"Vishn" <Vishn@xxxxxxxxxxxxxxxxxxxxxxxxxxxx> wrote in message news:AE498841-CBE0-4969-AE04-AE2EBE005387@xxxxxxxxxxxxxxxxxxxx

Hi.

WOW! That was quite fast. You are the best!!.

I ran the script, it executed succesfully in cmd prompt. Does it query the AD directly? How can we redirect the AD results to a text file(create a text file in the same directory as the vbs file & then).I

mean

the cmds which have to be added to the VBScript to redirect the output.

The

script looks like there is no input file required,so it should be querying

AD

directly when executed.Please correct me I am wrong. Is there a way where

in

it can accept input from a text file, run this script which will query the

AD

& redirect the results to an output text file?

Thank you so much for taking your time in helping me out with this issue.

Really appreciate your help!

Re: LDAP query information

"Richard Mueller" wrote:

Vishn wrote:

It takes the distinguishedName attribute but does not compute or display

in

the query result. Same with most of the fields of the "computer" object

attributes(Checked with most of the 'computer' object attributes in AD Schema).

Is there any way by which we can pull'

Canonical name of the object'

attribute for the 'Computer' object. maybe a VBScript if not a LDAP

query?

A VBScript which may check the computer accounts in a domain from AD

directly & compute the result with all the attributes of the 'Computer'

object or get the list of computers from a text file & then query AD &

compute the results.

Can you please construct a VBScript for me if thats the only solution?

That seems to be my last hope :(

Thanks for your help!

Hi.

The following VBScript program uses ADO to retrieve many of the

attributes

of all computer objects in the domain. It should be run at a command

prompt

using cscript. The values are output delimited by semicolons.

The output

can

be redirected to a text file and imported into a spreadsheet:

=====

Option Explicit

Dim objRootDSE, strDNSDomain, adoCommand,

adoConnection

Re: LDAP query information

Dim strBase, strFilter, strAttributes, strQuery, adoRecordset
Dim strNTName, lngUSN, arrCanonical, strCanonical
Dim strDN, strDesc, strRole, strManaged, strOS, strOSVer
Dim strSP, strLocation, strCreated, strChanged

' Determine DNS domain name.
Set objRootDSE = GetObject("LDAP://RootDSE)
strDNSDomain =
objRootDSE.Get("defaultNamingContext")

' Use ADO to search Active Directory.
Set adoCommand = CreateObject("ADODB.Command")
Set adoConnection = CreateObject("ADODB.Connection")
adoConnection.Provider = "ADsDSOObject"
adoConnection.Open "Active Directory Provider"
adoCommand.ActiveConnection = adoConnection

' Search entire domain.
strBase = "<LDAP:// & strDNSDomain & ">"

' Filter on computer object.
strFilter = "(objectCategory=computer)"

' Comma delimited list of attributes.
strAttributes =
"sAMAccountName.canonicalName.distinguishedName."
& "description.machineRole.managedBy.operatingSystem."
-
&
"location.operatingSystemVersion.operatingSystemServicePack."
-
& "whenCreated.whenChanged"

' Construct LDAP syntax query.
strQuery = strBase & ":" & strFilter & ":" & strAttributes &
":subtree"

' Execute the query.
adoCommand.CommandText = strQuery
adoCommand.Properties("Page Size") = 100
adoCommand.Properties("Timeout") = 30
adoCommand.Properties("Cache Results") = False
Set adoRecordset = adoCommand.Execute

' Enumerate the recordset and display values.
Do Until adoRecordset.EOF
strNTName =
adoRecordset.Fields("sAMAccountName").Value
arrCanonical =
adoRecordset.Fields("canonicalName").Value
strCanonical = arrCanonical(0)

Re: LDAP query information

```
strDN = adoRecordset.Fields("distinguishedName").Value  
strDesc = adoRecordset.Fields("description").Value  
strRole = adoRecordset.Fields("machineRole").Value  
strManaged = adoRecordset.Fields("managedBy").Value  
strLocation = adoRecordset.Fields("location").Value  
strOS = adoRecordset.Fields("operatingSystem").Value  
strOSVer =  
adoRecordset.Fields("operatingSystemVersion").Value  
strSP =  
adoRecordset.Fields("operatingSystemServicePack").Value  
strCreated = adoRecordset.Fields("whenCreated").Value  
strChanged = adoRecordset.Fields("whenChanged").Value  
Wscript.Echo strNTName & ";" & strCanonical & ";" &  
strDN  
& ";" & strDesc & ";" & strRole & ";" & strManaged  
& ";" & strLocation & ";" & strOS & ";" & strOSVer  
& ";" & strSP & ";" & strCreated & ";" & strChanged  
adoRecordset.MoveNext  
Loop  
adoRecordset.Close  
adoConnection.Close  
  
' Clean up.  
Set objRootDSE = Nothing  
Set adoCommand = Nothing  
Set adoConnection = Nothing  
Set adoRecordset = Nothing  
  
==  
Richard  
Microsoft MVP Scripting and ADSI  
Hilltop Lab – http://www.rlmuller.net
```