

Re: Copy Active directory Users to a SQL DB table – Daily

Source:

<http://www.tech-archive.net/Archive/Windows/microsoft.public.windows.server.scripting/2007-06/msg00119.html>

- *From:* "Al Dunbar" <AlanDrub@xxxxxxxxxxxxxxxxxxxxxx>
 - *Date:* Sat, 23 Jun 2007 11:11:23 -0600
-

"Richard Mueller [MVP]" <rlmueller-nospam@xxxxxxxxxxxxxxxxxxxxxx> wrote in message news:e%23K5L9StHHA.1184@xxxxxxxxxxxxxxxxxxxxxx

"Yas" <yasarl@xxxxxxxxxx> wrote in message news:1182556634.039711.251590@xxxxxxxxxxxxxxxxxxxxxx

On 21 Jun, 04:56, "Richard Mueller [MVP]" <rlmueller-nos...@xxxxxxxxxxxxxxxxxxxxxx> wrote:

Yas wrote:

I am trying to write a script using VBScript that gets a list of all active users in Active Directory including specified attributes 1,2,3..etc (if possible also specify which users to leave out eg.users in OU x) and then adds them to a table in a MS SQL database. Table will have corresponding columns..name, email,samAccountName,Attrib1,Attrib2 etc

And then daily scheduled runs of the script updates the SQL table from data in AD with any modifications made to the accounts and adds any new users that were not there before.

Is this possible using VBScript? sorry am restricted to only using

Re: Copy Active directory Users to a SQL DB table – Daily

vbscript. If so, any help or hints or ideas
would be greatly
appreciated.

Yes it can be done, using one ADO connection to retrieve the
information
from Active Directory and another to update the SQL Server
database
table.

The example below is not tested. I assume the same field
names in the

SQL

table as the attribute names in AD. The tricky part is
constructing the

T-SQL statements as strings, with all the commas, spaces,
and single

quotes

in the correct places. If there are errors, echo the T-SQL
statements so

you

can see them. It's usually a simple syntax mistake.

=====

Option Explicit

```
Dim adoADCommand, adoADConnection, strBase, strFilter,  
strAttributes
```

```
Dim objRootDSE, strDNSDomain, strQuery,  
adoADRecordset
```

```
Dim strConnect, adoSQLConnection, adoSQLCommand,  
adoSQLRecordset
```

```
Dim strNTName, strDN, strFirst, strMiddle, strLast,  
strDisplay
```

```
Dim arrDesc, strItem, strDesc
```

```
Dim intCount
```

```
' Setup ADO objects for Active Directory.
```

```
Set adoADCommand = CreateObject("ADODB.Command")
```

```
Set adoADConnection =
```

```
CreateObject("ADODB.Connection")
```

```
adoADConnection.Provider = "ADsDSOObject"
```

```
adoADConnection.Open "Active Directory Provider"
```

```
adoADCommand.ActiveConnection = adoADConnection
```

```
' Search entire Active Directory domain.
```

```
Set objRootDSE = GetObject("LDAP://RootDSE")
```

```
strDNSDomain =
```

```
objRootDSE.Get("defaultNamingContext")
```

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

```
' Filter on all user objects.
```

Re: Copy Active directory Users to a SQL DB table – Daily

```
strFilter = "(&(objectCategory=person)(objectClass=user))"  
  
' Comma delimited list of attribute values to retrieve.  
strAttributes =  
"sAMAccountName,distinguishedName,givenName."  
& "initials,sn,displayName,description"  
  
' Construct the LDAP syntax query.  
strQuery = strBase & ";" & strFilter & ";" & strAttributes &  
";subtree"  
adoADCommand.CommandText = strQuery  
adoADCommand.Properties("Page Size") = 100  
adoADCommand.Properties("Timeout") = 30  
adoADCommand.Properties("Cache Results") = False  
  
' Connection string for SQL database.  
' This assumes Windows integrated authentication.  
strConnect = "DRIVER=SQL Server:"  
& "Trusted Connection=Yes;"  
& "DATABASE=MyDatabase:"  
& "SERVER=MyServer"  
  
' Connect to SQL Server database.  
Set adoSQLConnection =  
CreateObject("ADODB.Connection")  
adoSQLConnection.ConnectionString = strConnect  
  
' Create Recordset object for SQL Server database.  
Set adoSQLRecordset = CreateObject("ADODB.Recordset")  
adoSQLRecordset.ActiveConnection = adoSQLConnection  
  
' Create Command object for SQL Server database.  
Set adoSQLCommand =  
CreateObject("ADODB.Command")  
adoSQLCommand.ActiveConnection = adoSQLConnection  
  
' Run the query on Active Directory.  
Set adoADRecordset = adoADCommand.Execute  
  
' Enumerate the resulting recordset.  
Do Until adoADRecordset.EOF  
' Retrieve values for users in Active Directory.  
strNTName =  
adoADRecordset.Fields("sAMAccountName").Value  
strDN =  
adoADRecordset.Fields("distinguishedName").Value  
strFirst = adoADRecordset.Fields("givenName").Value  
strMiddle = adoADRecordset.Fields("initials").Value  
strLast = adoADRecordset.Fields("sn").Value  
strDisplay = adoADRecordset.Fields("displayName").Value  
' The description field is multi-valued, although there
```

Re: Copy Active directory Users to a SQL DB table – Daily

```
' can never be more than one value. ADO retrieves the  
' value as either Null or an array of one string value.  
arrDesc = adoADRecordset.Fields("description").Value  
If IsNull(arrDesc) Then  
strDesc = ""  
Else  
For Each strItem In arrDesc  
strDesc = strItem  
Next  
End If  
' Check if user exists in table ADUsers of SQL database.  
adoSQLRecordset.Source = "SELECT COUNT(*) AS  
NumRecords "  
& "FROM dbo.ADUsers "  
& "WHERE sAMAccountName = " & strNTName & ""  
adoSQLRecordset.Open  
intCount =  
CInt(adoSQLRecordset.Fields("NumRecords").Value)  
  
If (intCount = 0) Then  
' User is new, add to SQL table.  
adoSQLCommand.CommandText = "INSERT INTO  
dbo.ADUsers "  
& "(sAMAccountName, distinguishedName, givenName, "  
& "initials, sn, displayName, description) "  
& "VALUES(" & strNTName & ", " & strDN & ", "  
& strFirst & ", " & strMiddle & ", " & strLast  
& ", " & strDisplay & ", " & strDesc & ")  
Else  
' User already in SQL table, update the existing record.  
adoSQLCommand.CommandText = "UPDATE  
dbo.ADUsers "  
& "SET distinguishedName = " & strDN & ", "  
& "givenName = " & strFirst & ", "  
& "initials = " & strMiddle & ", "  
& "sn = " & strLast & ", "  
& "displayName = " & strDisplay & ", "  
& "description = " & strDesc & " "  
& "WHERE sAMAccountName = " & strNTName & ""  
End If  
' Run the T-SQL statement on the SQL database.  
adoSQLCommand.Execute  
adoSQLRecordset.Close  
  
' Move to the next record in the recordset of AD users.  
adoADRecordset.MoveNext  
Loop  
  
' Clean up.  
adoADRecordset.Close  
adoADConnection.Close
```

Re: Copy Active directory Users to a SQL DB table – Daily

adoSqlConnection.Close

This is great, thank you! :-) I modified it a bit to suite my needs and it works well.

I was wondering does anyone know of a way to exclude certain OUs when searching Active directory? I've used UserAccountControl to exclude disabled users but wouldlike to exclude certain OUs ie. the ones that contain system Admins etc not being able to find anything ont he Net that works.

Thanks

Yas

There's no way to either include or exclude OU's in the filter. This is because there is no attribute of AD objects for the "Parent" container. You would have to parse the distinguishedName (DN), but you can only specify full distinguishedNames (no wildcards allowed with any DN attributes).

The solution is to return all user objects, then in the loop where you enumerate the recordset, parse the DN of each and exclude those in the OU's you want to skip. Even here, the best way to tell the OU is to bind to the user object (with the DN) and use the Parent method of the object, which returns the DN of the parent OU/container. Unfortunately, Parent is a property method, not an attribute, so ADO cannot retrieve it.

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

There might be other options depending on the OU structure. If, for example, all regular users were contained in OU's nested within a custom userOU in the root, and all admin users in a custom adminOU in the root, a search done against userOU would necessarily exclude administrators. Our structure runs along these lines, and the reason is to avoid delegating admin access over administrative accounts too widely.

Also, if the real reason for excluding OU's containing administrative accounts is to exclude all (and only) administrative accounts, it might be more accurate to do as Richard suggests, but exclude only accounts that can be identified as having admin rights. That can get a little tricky unless you have a good understanding of what the differences are between a non-privileged user account, and an admin account. If admin means domain admin, this is relatively easy; if it also includes OU admins, well...

Re: Copy Active directory Users to a SQL DB table – Daily

Re: Copy Active directory Users to a SQL DB table – Daily

/AI

.