

# Re: Script to change passwords for multiple users to logon name

---

*Source:*

<http://www.tech-archive.net/Archive/Scripting/microsoft.public.scripting.vbscript/2008-02/msg00273.html>

---

- *From:* Brad <[Brad@xxxxxxxxxxxxxxxxxxxxxxxxxxxxx](mailto:Brad@xxxxxxxxxxxxxxxxxxxxxxxxxxxxx)>
  - *Date:* Sun, 10 Feb 2008 21:39:01 -0800
- 

Hi Richard,

Thank you for your response and it looks like this will do what I need however when I try to run the set password script I get an error that states "Password NOT set for: " message. Any ideas what would cause this error?

Thanks again for your help.

"Richard Mueller [MVP]" wrote:

Brad wrote;

Is there a script or an easy way to reset all passwords for all users in a specified OU in an AD domain to there logon name? Inside this OU there are child OUs that contain users also that need there password reset to the logon name. I have found a script to reset it to the display name but not to the logon name. Any help or info would be greatly appreciated.

Rather than blindly modifying all users, I like to have a script read names from a spreadsheet. This way I know exactly which users are affected and how (what the new password will be). I have an example VBScript program that resets passwords for users specified in an Excel spreadsheet linked here:

<http://www.rlmueller.net/Set%20Passwords%203.htm>

The user Distinguished Name is in the first column and the new password in the second column. The page has a link to a program that will create the spreadsheet with Distinguished Names of all users in the domain, so you only need add the passwords. That program is linked here:

Re: Script to change passwords for multiple users to logon name

<http://www.rlmueller.net/Create%20User%20List%203.htm>

You can modify this program to just document the users in an OU (and any child OU's) by changing the base of the ADO query. In place of strDNSDomain, use the Distinguished Name of the OU. This statement:

```
strQuery = "<LDAP:// & strDNSDomain & ">:" & strFilter  
& "":distinguishedName:subtree"
```

would become similar to:

```
strQuery = "<LDAP://ou=West,dc=MyDomain,dc=com>:" & strFilter  
& "":distinguishedName:subtree"
```

If you don't want to fill in the new passwords in the second column, you could also have this program retrieve the value of the sAMAccountName attribute (the NT name of the user, also called the "pre-Windows 2000 logon name") and document this in the second column of the spreadsheet. To do this, you would add the sAMAccountName attribute to the comma delimited list of attribute values to retrieve. Then the query statement above becomes:

```
strQuery = "<LDAP://ou=West,dc=MyDomain,dc=com>:" & strFilter  
& "":distinguishedName,sAMAccountName:subtree"
```

Then in the loop where the resulting recordset is processed:

```
=====  
Dim strName  
Do Until adoRecordset.EOF  
strDN = adoRecordset.Fields("distinguishedName").Value  
' Escape any forward slash characters, "/", with the backslash  
' escape character. All other characters that should be escaped are.  
strDN = Replace(strDN, "/", "\")  
strName = adoRecordset.Fields("sAMAccountName").Value  
objSheet.Cells(k, 1).Value = strDN  
objSheet.Cells(k, 2).Value = strName  
k = k + 1  
adoRecordset.MoveNext  
Loop
```

I added the Dim statement for the new variable strName because the script uses "Option Explicit". Once the spreadsheet is created, you can review it make sure the correct users are listed. You can also verify the new passwords. Then run the first program, SetPWForUserList3.vbs. Note that this program also enables the accounts and expires the passwords so the users must change them the next time they logon.

```
==  
Richard Mueller  
Microsoft MVP Scripting and ADSI  
Hilltop Lab – http://www.rlmueller.net  
==
```

Re: Script to change passwords for multiple users to logon name