Dmitry’s PowerBlog: PowerShell and beyond

Dmitry Sotnikov’s view on PowerShell, PowerGUI and everything he sees around

Blog

About

« MVP Award RenewedTerminal Services PowerPack »

Update AD from CSV

Published October 3, 2008 Active Directory , AD , AD cmdlets , cmdlets , Examples , one-liner , oneliner , PowerShell 27 Comments

Suppose you have a CSV file (a text file with columns separated by commas) with the properties for AD user accounts you want to update. How do you do this in PowerShell?

Turns out, that we talked a lot about creating new accounts from CSV files before, but not about updating existing ones. Let’s fix this right away.

I will be using AD user accounts in my examples, but it is fairly easy to adapt them to other AD objects: groups, computers, OUs, DNS records, and so on.

The command actually depends on the CSV you get. The easiest case is when the column names are exactly the same as Set-QADUser parameters. For example, let’s say you have a CSV file in which you have a samAccountName column which you want to use to locate the accounts to update and Title and Department columns with the new values to set:

samAccountName,Title,Department

test1,Manager,Accounting

test2,Developer,RD

test3,SC,Pre-Sales

The onliner to apply this file to your AD is as simple as:

Import-Csv c:\update.csv | Set-QADUser -Identity { $\_.samAccountName }

You basically pipe import into Set-QADUser and specify which column to use as the identity anchor.

Easy!

Now, suppose that life is not so easy and either you do not control the column labels or you need to update attributes which either do not match the parameter names or have no matching parameters at all. Like:

samAccountName,Job,ExtensionAttribute1,ExtensionAttribute2

test1,Manager,M,Yes

test2,Developer,S,No

test3,SC,XXL,Maybe

The automated column matching will not work here but we can use ForEach-Object loop and match the parameters manually + use ObjectAttributes for attributes with no parameters:

Import-Csv c:\update.csv | ForEach-Object {

Set-QADUser $\_.samAccountName -Title $\_.Job `

-ObjectAttributes @{ExtensionAttribute1=($\_.ExtensionAttribute1);

ExtensionAttribute2=($\_.ExtensionAttribute2)}

}

Now we can update from CSV any account properties we want!

Tags: AD, AD cmdlets, Active Directory, Examples, PowerShell, cmdlets, one-liner, oneliner

Add to: | Technorati | Digg | del.icio.us | Yahoo | BlinkList | Spurl | reddit | Furl |

About these ads

Share this post!

Email

Facebook

Twitter

More

Like this:

Like

Loading...

27 Responses to “Update AD from CSV”

Feed for this Entry Trackback Address

Rich Harmer

October 6, 2008 at 7:44 pm

Can’t this be slightly modified to create users too?

Reply

Dmitry Sotnikov

October 7, 2008 at 11:28 am

Rich,

Yes, you can use a similar approach with the New-QADUser cmdlet:

http://dmitrysotnikov.wordpress.com/2008/01/21/ad-user-provisioning-from-csv-got-easier/

Dmitry

Reply

Anuar

June 25, 2009 at 1:02 pm

How to deal with attribute that has dash ‘-’ in it?

I got an error: “Missing ‘=’ operator after key in hash literal.”

Reply

Dmitry Sotnikov

June 25, 2009 at 1:22 pm

Anuar,

Quotation marks should help. E.g.: @{ ‘attribute-with-dash’ = 25 }

If I got the question wrong or you have more issues there please post your questions to our AD PowerShell forum at: http://powergui.org/forum.jspa?forumID=173

Dmitry

Reply

Ozk4r

February 3, 2011 at 1:24 am

How i run this update from a schedule task?

Thanks

Reply

Dmitry Sotnikov

February 3, 2011 at 6:00 am

Scheduling a script as task is easy:

1. Make sure that the script has all the snapins/modules it needs. E.g. this one needs the QAD cmdlets so you might want to add this to the script:

Add-PSSnapin Quest\*

2. Run PowerShell as administrator and set execution policy to allow local script execution:

Set-ExecutionPolicy RemoteSigned

3. Schedule powershell.exe running your script. E.g.:

c:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe -NoProfile -Noninteractive -command “. c:\scripts\update-ad.ps1″

(See this post for another example of script scheduling: http://dmitrysotnikov.wordpress.com/2008/05/29/monitor-web-site-availability/)

Reply

Dmitry Sotnikov

February 3, 2011 at 9:14 pm

Here’s a detailed post on scheduling PowerShell scripts: http://dmitrysotnikov.wordpress.com/2011/02/03/how-to-schedule-a-powershell-script/

Oscar

February 9, 2011 at 12:23 am

Excelent, Thank you!

Reply

Kirby

June 21, 2011 at 6:25 pm

If your cvs file just contains old phonenumbers and new phonenumbers can you use the old phonenumber for the identity and then replace it with the new phonenumber?

Reply

Dmitry Sotnikov

June 21, 2011 at 9:58 pm

Kirby,

Yes, you can, just use Get-QADUser -PhoneNumber (http://wiki.powergui.org/index.php/Get-QADUser)

Your script will probably look like:

import-csv c:\numbers.csv | foreach-object {

Get-QADUser -PhoneNumber $\_.OldNumber | Set-QADUser -PhoneNumber $\_.NewNumber

}

If you get stuck or have additional questions, ask those at our AD and PowerShell forum here: http://powergui.org/forumindex.jspa?categoryID=55

Dmitry

Reply

Kirby

June 22, 2011 at 4:51 pm

Hi Dmitry,

Thanks for the help, it worked great. I added a small peice so I could run it agaist an targeted OU. Thanks again

$OuDomain = “OU=The,OU=One,OU=You,OU=Want,DC=mydomain,DC=com”

import-csv c:\scriptest\phonenumbers.csv | foreach-object { Get-QADUser -SearchRoot $OuDomain -PhoneNumber $\_.OldNumber | Set-QADUser -PhoneNumber $\_.NewNumber }

Reply

Annie

February 22, 2012 at 6:48 pm

Hello, Your script works well but it still searches all users in AD instead of the ones in that OU. Is there a way to be sure that it is targeting only one OU?

Reply

Dmitry Sotnikov

February 22, 2012 at 8:30 pm

Yes, just use -SearchRoot parameter. See http://wiki.powergui.org/index.php/Get-QADUser

Annie

February 23, 2012 at 2:04 pm

I have used the searchroot command but it still searches the entire directory. Please see below.

Get-QADUser -Searchroot “OU=Guests,OU=People,DC=XXX,DC=XXX” -id $\_.user

Dmitry Sotnikov

February 23, 2012 at 5:31 pm

This is very weird. Can you try posting the issue to the PowerShell and AD forum at http://powergui.org?

Vladimir Valchev

September 11, 2011 at 10:21 pm

Thank you Dmitry, saved my day

Reply

Dmitry Sotnikov

September 12, 2011 at 6:23 pm

My pleasure!

Reply

Michael

November 14, 2011 at 3:03 pm

I have used the Powershell Addin. It ist very usefull for me.

But I have one problem. I have a csv file, but not every entry in the csv file have an user account. Everytime when the script not found the user, the script is stop the working.

Could help me to solve this problem?

Many thanks

Reply

Dmitry Sotnikov

November 14, 2011 at 3:22 pm

Michael,

One workaround would be to use Test-QADObject as described here: http://dmitrysotnikov.wordpress.com/2010/03/22/test-if-ad-object-path-exists/

Dmitry

Reply

Allan

February 24, 2012 at 1:02 am

Hi Dmitry,

Am a novice Powershell user.

Have tried to use your script and this does not appear to work for me.

Not sure if it’s an environmental issue or something I am doing wrong.

Also hard to see what’s happening as the script is not returning any errors – so it appears that this is working.

Am trying to run the following script in a ps1 file: -

Import-Csv “C:\Temp\update\_users.csv” | Set-ADUser -server “inf003.org.local” -Identity { $\_.samAccountName } 2>> “C:\Temp\User\_Error\_Log.txt”

Script runs ok (as in no errors reported) however does not update the user in the CSV file.

CSV file is as follows: -

samAccountName,Title,EmployeeNumber

user1,Testing123,567890

So as you can see all CSV titles are correct and I think it should work?

The changes I made to your original script at the top of this post was the server name (as reading other posts on the web I was coming up against a problem where I did not have sufficient rights to run the commands, even though I was logged in with DA privilages and running the console as Admin) and I am attempting to log any errors to a log file using 2>>

Any thoughts?

Help much appreciated!

Dmitry Sotnikov

February 24, 2012 at 3:20 am