Managing user accounts with Windows PowerShell

Creating and managing user accounts is a common Active Directory administration task. Windows PowerShell provides considerable flexibility in how this can be done in Windows Server 2012 and Windows Server 2012 R2. Typing Get-Command *ADUser at a Windows PowerShell prompt shows there are four cmdlets for managing user accounts:

- **New-ADUser** - Creates a new Active Directory user
- **Get-ADUser** - Gets one or more Active Directory users so that you can perform some action with them
- **Set-ADUser** - Modifies the properties of an existing Active Directory user
- **Remove-ADUser** - Removes the specified user from Active Directory

Any administration of user accounts using Windows PowerShell involves using one or more of these cmdlets. The following sections demonstrate some of the ways you can create new user accounts using the New-ADUser cmdlet. The approach you choose depends on the particular needs of your situation.

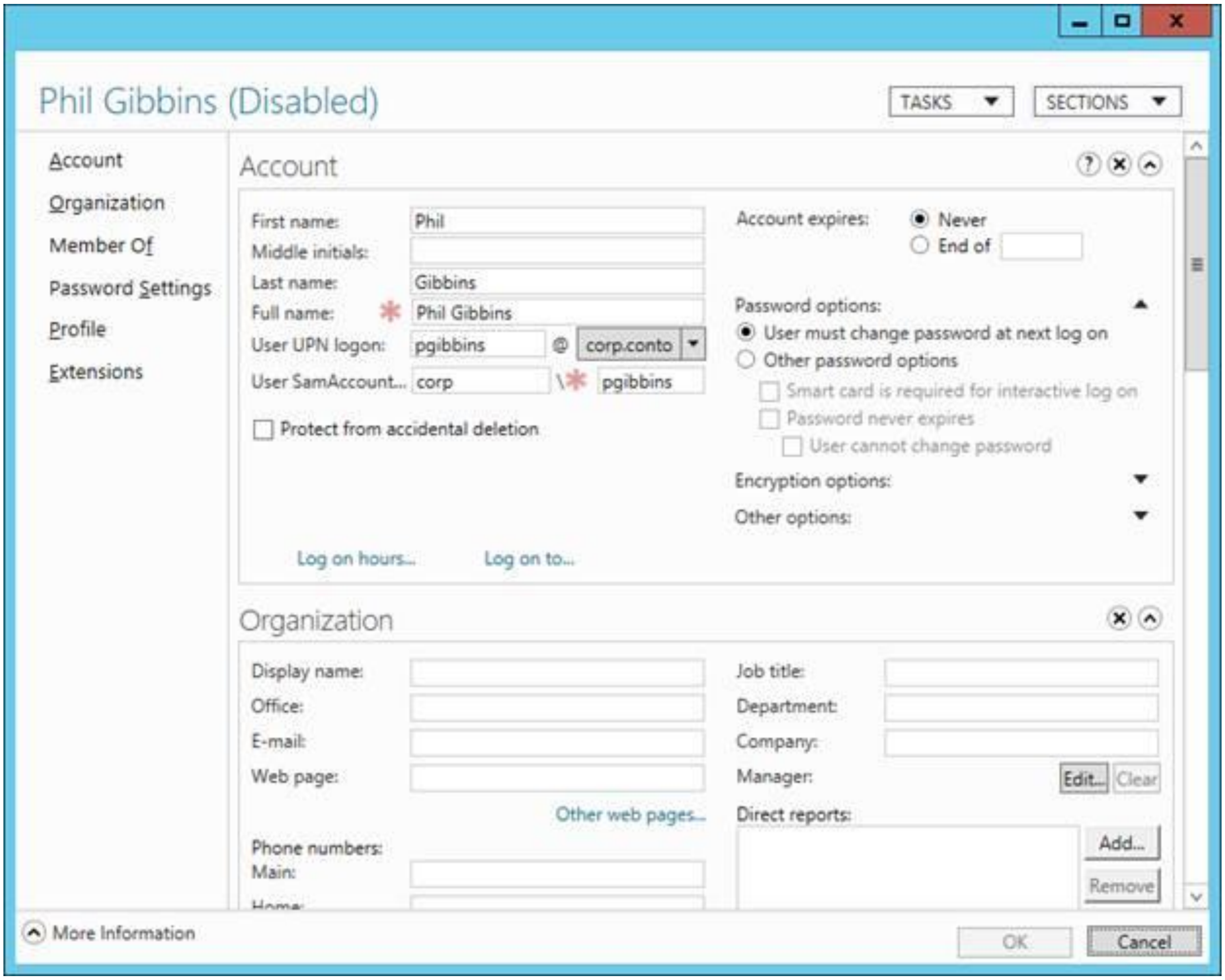
Example 1: Create a single new user account

To create a new user account for Phil Gibbins using pgibbins for the user's SAM account name and pgibbins@corp.contoso.com for the user's UPN logon, you can use the New-ADUser cmdlet as follows:

```
PS C:\> New-ADUser -Name "Phil Gibbins" -GivenName Phil -Surname Gibbins `

-SamAccountName pgibbins -UserPrincipalName pgibbins@corp.contoso.com
```

Note that there is no output if the command runs successfully. The resulting properties of the new user account when it is opened in ADAC are shown in Figure 1. Note that there are numerous other properties you could have specified when creating the account. Each of these additional properties has a parameter associated with it when using the New-ADUser cmdlet.



Phil Gibbins (Disabled)

Account

First name: Phil
Middle initials:
Last name: Gibbins
Full name: Phil Gibbins
User UPN logon: pgibbins @ corp.conto
User SamAccount... corp \ pgibbins

□ Protect from accidental deletion

Account expires: ● Never ○ End of

Password options:
● User must change password at next log on
○ Other password options
□ Smart card is required for interactive log on
□ Password never expires
□ User cannot change password

Figure 1: Create a new user account using the New-ADUser cmdlet.

Note that if you try the preceding example, you'll discover that the user account for Phil Gibbins is created in the Users container of the domain. To create a user account in a different location, you must specify the -Path parameter with this command. For example, to create this account in the location ou=Seattle Users OU,ou=Seattle OU OU,dc=corp,dc=contoso,dc=com in Active Directory, you could append -Path "ou=Seattle Users OU,ou=Seattle OU OU,dc=corp,dc=contoso,dc=com" to the command used in the preceding example.

Example 2: Create a new user account and specify a password

To specify a password when you create the user account for Phil Gibbins, you can use the Read-Host cmdlet. With this cmdlet, you enter a password when you run the command, as shown by the highlighted code in the following example:

```
PS C:\> New-ADUser -Name "Phil Gibbins" -GivenName Phil -Surname Gibbins `
```

```
-SamAccountName pgibbins -UserPrincipalName pgibbins@corp.contoso.com `
-AccountPassword (Read-Host -AsSecureString "AccountPassword")
```

Example 3: Create and enable a new user account

When you use the New-ADUser cmdlet to create a user account, the new account is disabled and cannot be enabled unless either of the following has occurred:

- A valid password has been set for the account.
- The -PasswordNotRequired parameter has been set to true.

To create a user account for Phil Gibbins, specify a password, and enable the new account, you can use the following command:

```
PS C:\> New-ADUser -Name "Phil Gibbins" -GivenName Phil -Surname Gibbins `
-SamAccountName pgibbins -UserPrincipalName pgibbins@corp.contoso.com `
-AccountPassword (Read-Host -AsSecureString "AccountPassword") `
-PassThru | Enable-ADAccount
```

The -PassThru parameter, which has been added to the New-ADUser command just shown, returns the newly created user account object so that it can be piped into the Enable-ADAccount cmdlet to enable the new account.

Example 4: Bulk-create new user accounts

A good example of how you can use Windows PowerShell to automate a common Active Directory management task is the bulk creation of users. For example, you can combine the previous examples with the Import-Csv cmdlet, which enables you to read in data from a comma-separated values (CSV) file to create multiple user accounts in a single operation.

To illustrate this, the file new-users.csv contains a line of header information followed by attributes for three user accounts as follows:

```
Name,GivenName,Surname,SamAccountName,UserPrincipalName
Arno Bost,Arno,Bost,abost,abost@corp.contoso.com
Peter Fischer,Peter,Fischer,pfischer,pfischer@corp.contoso.com
Manish Chopra,Manish,Chopra,mchopra,mchopra@corp.contoso.com
```

The following command first reads the CSV file and pipes its contents into the New-ADUser cmdlet, then sets the password for each user account as Pa$$w0rd, and finally enables the accounts:

```
PS C:\> Import-Csv C:\data\new-users.csv | New-ADUser -PassThru | `

Set-ADAccountPassword -Reset `

-NewPassword (ConvertTo-SecureString -AsPlainText 'Pa$$w0rd' -Force) `

-PassThru | Enable-ADAccount
```

The highlighted portion of this command takes the string "Pa$$w0rd" and converts it from plain text to a secure string so that it can be used by the -NewPassword parameter of the Set-ADAccountPassword cmdlet. The -Force parameter is needed to suppress the confirmation prompt generated by use of the -AsPlainText parameter.

Bulk creation of user accounts, computer accounts, groups, and other types of directory objects involves two steps:

- Creating the source file with the information for the accounts that need to be created
- Creating the command or script that takes the source file and uses it to bulk-create the new accounts

The CSV format used in the example in this section is a universal format supported by numerous applications, including Microsoft Excel, Microsoft Access, and even Microsoft SQL Server. By using a program like Excel to create the source information and save it in CSV format, you can quickly and easily bulk-create accounts in Active Directory.

Example 5: Create new user accounts from a template account

A template account is an account you use as a basis for creating other accounts. By configuring template account properties that are common to the other accounts you need to create, you can reduce the amount of information you need to provide for creating the additional accounts.

For example, you could configure properties like the account expiration date and password options in a template account if these will be the same for the other user accounts you need to create. You may also configure properties like Company, Address, City, and Country in the template account. By doing this, you won't need to specify these properties when you create the other user accounts.

TIP:
One of the practice exercises at the end of Chapter 5 of my book Training Guide: Installing and

Configuring Windows Server 2012 R2 demonstrates how you can create a template account and then use it as a basis for creating additional user accounts.

Some Additional Tips

Finally, here a couple additional tips I've gleaned from my colleagues in IT and from readers of our newsletter WServerNews.com.

Copying user account properties

You can copy the properties of one user account to create another. In effect you are using the first user account as a template for creating the second one. But you need to be careful when you try to do this. For example, this command works:

```
$user=Get-ADUser -Identity bobsmith

New-ADUser -Instance $user -SamAccountName sarajones
```

But this command fails:

```
$user=Get-ADUser -Identity bobsmith -Properties *

New-ADUser -Instance $user -SamAccountName sarajones
```

The reason that the second command fails is because -Properties * returns all properties of the user account object and some of those properties like logonCount and badPwdCount can only be modified by the Security Accounts Manager (SAM) and not from PowerShell.

Creating user accounts for lab testing

Do you need to create a large number of user accounts in Active Directory in your lab environment for testing purposes? Here's a simple script a colleague forwarded to me that creates 5,000,000 users in the Chicago OU of the Contoso.com domain:

```
for($i=1; $i -le (5*[math]::pow(10,6)); $i++) {

New-ADUser -SamAccountName "Bob$i" -Name "Bob$i" -GivenName "Bob$i" `

-Surname "Smith" -DisplayName "Bob$i Smith" -Path 'OU=Chicago,DC=contoso,DC=com'

}
```