

PowerShell ADS User Script

Time required: 30 minutes

- Comment each line of code as shown in the tutorials and other code examples.
- Follow all directions carefully and accurately.
- Think of the directions as minimum requirements.

This script will demonstrate how to search for a user in Active Directory by their username, modify and display user properties, reset their password, and add the user to a specific group.

Requirements

- Search and find a user in Active Directory by their username
- Modify the DisplayName and EmailAddress for a user
- Display the user's DisplayName and EmailAddress
- Reset the user's password
- Add the user to a specific AD group

Tutorial

```
1 # Filename: PowerShellUserScript.ps1
2
3 # Import the Active Directory module, allowing us to use AD-related cmdlets
4 Import-Module ActiveDirectory
```

Import the Active Directory module. This module provides cmdlets (commands) for working with Active Directory in PowerShell.

```
6 # Set the Username variable to the user we want to search for
7 # in Active Directory, in this example, "sales1"
8 $Username = "sales1"
```

```

10 # Get-ADUser is a cmdlet that searches for a user in Active Directory
11 # based on the filter condition.
12 # The filter {SamAccountName -eq $Username} searches for
13 # a user with the specified username.
14 # We specify the DisplayName and EmailAddress property to be retrieved.
15 $User = Get-ADUser `
16     -Filter { SamAccountName -eq $Username } `
17     -Properties DisplayName, EmailAddress

```

Set the username we want to search for in Active Directory, in this case, **sales1**. We use the **Get-ADUser** cmdlet to search for the user based on their username. We also specify that we want to retrieve the **DisplayName** and **EmailAddress** property.



The ``` backtick character allows us to break the command into multiple lines.

```

19 # Check if the $User variable is not equal to null, meaning a user was found
20 if ($User -ne $null) {
21     # Confirm that the username was found
22     Write-Host "User $Username found."

```

Is the user found in Active Directory? If the user exists, we proceed with further actions. If the user is not found, we handle that scenario accordingly later in the program with an **else** statement.

```

24 # Create variables to store the new DisplayName and Email Address
25 $NewDisplayName = "Joe Sales"
26 $NewEmailAddress = "joe.sales@mybusiness.com"

```

Define new values for the **DisplayName** and **EmailAddress**.

```

28 # Modify the user's DisplayName and Email address
29 Set-ADUser -Identity $User `
30 -DisplayName $NewDisplayName `
31 -EmailAddress $NewEmailAddress

```

Use the **Set-ADUser** cmdlet to update the user's attributes.

```

36 # Display the user's DisplayName and EmailAddress
37 # $User.Name and $User.EmailAddress are properties
38 # of the user object returned by Get-ADUser
39 Write-Host "User Full Name: $($User.DisplayName)"
40 Write-Host "User Email: $($User.EmailAddress)"
41 # You can add more properties as needed

```

Use **Write-Host** to display the user's Display Name and Email Address. The **DisplayName** and **EmailAddress** properties are accessed as **\$User.DisplayName** and **\$User.EmailAddress**.

```

43 # Create a secure string for the new password
44 $NewPassword = ConvertTo-SecureString `
45 -String "Password01" `
46 -AsPlainText `
47 -Force

```

Create a secure string for the new password.

```

49 # Use Set-ADAccountPassword to change the user's password.
50 Set-ADAccountPassword -Identity $User -NewPassword $NewPassword

```

Use the **Set-ADAccountPassword** cmdlet to change the user's password.

```

52 # Confirm password reset
53 Write-Host "Password reset successfully."

```

Confirm password reset.

```

55 # Use Add-ADGroupMember to add user to specific group
56 $GroupName = "G_Accounting"
57 Add-ADGroupMember -Identity $GroupName -Members $User
58
59 # Confirm that the user was added to the group
60 Write-Host "User added to $GroupName."
61 }

```

Add the user to a specific group.

```
62 else {  
63     # If the user was not found in Active Directory  
64     # display a message indicating that the user was not found  
65     Write-Host "User $Username not found."  
66 }
```

Example run:

```
PS C:\Users\Administrator> C:\Users\Admi  
User sales1 found.  
Display Name and Email Address updated.  
User Full Name: Joe Sales  
User Email: joe.sales@mybusiness.com  
Password reset successfully.  
User added to G_Accounting.
```

Assignment Submission

1. Attach the program files.
2. Attach screenshots showing the successful operation of the programs.
3. Submit in Blackboard.