# Week 6 PowerShell

Contents

[Week 6 PowerShell 1](#_Toc209466613)

[Tutorial 1: WMI 1](#_Toc209466614)

[Tutorial 2: Process Query Script 2](#_Toc209466615)

[Tutorial 3: PowerShell Cmdlet Output 4](#_Toc209466616)

[Tutorial 4: Manage Active Directory Users 6](#_Toc209466617)

[Assignment Submission 7](#_Toc209466618)

Time required: 60 minutes

# Tutorial 1: WMI

You can use a physical machine or a Virtual Machine for this lab.

In this Hands-On Project, you query WMI using Windows PowerShell.

1. Right-click the Start menu and choose Windows PowerShell (Admin) to open Windows PowerShell.
2. At the prompt, type **alias gwmi** and press enter. Note that gwmi is an alias to Get-WmiObject.
3. Execute the following commands at the command prompt, in turn. For each one, interpret the output (referencing the aliases and cmdlets within this module, as necessary).

|  |
| --- |
| gwmi win32\_bios |
| Example: Retrieves BIOS information |
| gwmi win32\_processor |
| Click or tap here to enter text. |
| gwmi win32\_processor | gm |
| Click or tap here to enter text. |
| gwmi win32\_computersystem |
| Click or tap here to enter text. |
| gwmi win32\_computersystem | fl \* |
| Click or tap here to enter text. |
| gwmi win32\_logicaldisk |
| Click or tap here to enter text. |
| gwmi win32\_logicaldisk | fl \* |
| Click or tap here to enter text. |
| gwmi win32\_diskdrive |
| Click or tap here to enter text. |
| gwmi win32\_diskdrive | fl \* |
| Click or tap here to enter text. |
| gwmi win32\_share |
| Click or tap here to enter text. |
| gwmi win32\_share | fl \* |
| Click or tap here to enter text. |
| gwmi win32\_networkadapterconfiguration |
| Click or tap here to enter text. |
| gwmi win32\_networkadapterconfiguration | fl \* |
| Click or tap here to enter text. |
| gwmi win32\_desktop |
| Click or tap here to enter text. |

Close Windows PowerShell.

# Tutorial 2: Process Query Script

Use Server 2022.

In this Hands-On Project, you create and execute a basic PowerShell script using Windows PowerShell ISE, and execute it on the system.

1. Click Start. Next, right-click Windows PowerShell ISE and click More, Run as administrator.
2. Click the New Script button above the Windows PowerShell pane, and enter the following contents.
3. Click File, Save As 🡪 **query\_process.ps1**

|  |
| --- |
| # Filename: process\_query.ps1  # This script prints process information to the screen for a  # process that the user is prompted to supply during script execution  # Read process name into the $ans variable  $ans = Read-Host "What process would you like to query?"  # Query the process using the value in the $ans variable  # This command is all on one line  Get-WmiObject win32\_process -Filter "name='$ans'" | Format-Table HandleCount, VirtualSize, UserModeTime, KernelModeTime, ProcessID, Name |

1. Click the Run Script button (or press F5) to test your script in the Windows PowerShell console.
2. Type the value **svchost.exe** and press enter when prompted.Note If there are errors in your script, fix the typos in your script.
3. Close Windows PowerShell ISE.
4. Right-click the Start menu and choose Windows PowerShell (Admin) to open Windows PowerShell.
5. At the prompt, type cd \ and press enter to switch to the root of C:\.
6. type **.\query\_process.ps1** and press enter to execute your script again. Type the value svchost.exe and press enter when prompted.
7. Insert a screenshot:

Click or tap here to enter text.

# Tutorial 3: PowerShell Cmdlet Output

Use Server 2022.

In this Hands-On Project, you modify the output of Windows PowerShell cmdlets.

1. At the prompt, type **cd \** and press enter to switch to the root directory. (There is a space between cd and \)
2. Type **dir** and press enter to list the contents of this directory.
3. At the prompt, type **alias dir** and press enter.
4. Type **alias gci** and press enter. Note that dir and gci are aliases to the Get-ChildItem cmdlet.
5. **Insert a screenshot:**

Click or tap here to enter text.

1. Execute the following commands at the command prompt, in turn. For each one, interpret/explain the output (referencing the aliases and cmdlets within this module, as necessary).
2. **Get-ChildItem** (For example, Get-ChildItem is an alias for dir)

Click or tap here to enter text.

1. **gci**

Click or tap here to enter text.

1. **gci | sort -property name**

Click or tap here to enter text.

1. **gci | Format-List**

Click or tap here to enter text.

1. **gci | Format-List -property name, lastwritetime**

Click or tap here to enter text.

1. **gci | Format-Wide**

Click or tap here to enter text.

1. **gci | Format-Wide –column 3**

Click or tap here to enter text.

1. **gci | Format-Wide –column 3 –property length**

Click or tap here to enter text.

1. **gci | Format-Wide –column 3 –property name –groupby length**

Click or tap here to enter text.

1. **gci | Format-Table**

Click or tap here to enter text.

1. **gci | Format-Table –property name, length, lastwritetime**

Click or tap here to enter text.

1. **gci -recurse**

Click or tap here to enter text.

1. **gci –recurse –include \*.txt**

Click or tap here to enter text.

Other cmdlets that generate a large amount of information may require that you use additional Windows PowerShell features to modify command output for organization and readability.

1. Execute the following commands at the command prompt, in turn. For each one, interpret/explain the output (referencing the aliases and cmdlets within this module, as necessary).
2. **Get-EventLog System | more**

Click or tap here to enter text.

1. **Get-EventLog System | Group-Object eventid | more**

Click or tap here to enter text.

1. **Get-EventLog System | Group-Object eventid | Out-GridView**  
   (Close the GridView window when finished.)

Click or tap here to enter text.

1. **Get-EventLog System | Group-Object eventid | ogvse**  
   (Close the GridView window when finished.)

Click or tap here to enter text.

1. **Get-Process | ogv**  
   (Close the GridView window when finished.)

Click or tap here to enter text.

1. **Get-Process | ConvertTo-HTML | Out-File C:\PList.html**  
   **Invoke-Item C:\PList.html**  
   (This should open in your default web browser.)

Click or tap here to enter text.

1. **Get-Process | Export-CSV C:\PList.csv**  
   **Invoke-Item C:\PList.csv**  
   (This should open in Excel.)

Click or tap here to enter text.

# Tutorial 4: Manage Active Directory Users

Use Server 2025.

1. Connect to Active Directory.

|  |
| --- |
| Import-Module ActiveDirectory |

1. To create a new user, use the **New-ADUser** cmdlet. This command is all one line.

|  |
| --- |
| New-ADUser -Name "John Doe" -SamAccountName "johndoe" -UserPrincipalName "johndoe@mydomain.com" -Path "OU=Employees,DC=mydomain,DC=local" -AccountPassword (ConvertTo-SecureString "Password01" -AsPlainText -Force) -Enabled $true |

1. Insert a screenshot of Active directory showing the new user.

Click or tap here to enter text.

1. Retrieve user information with **Get-ADUser**.

|  |
| --- |
| Get-ADUser -Identity "johndoe" -Properties \* |

1. Insert a screenshot of the results.

Click or tap here to enter text.

1. Use **Set-ADUser** to update user attributes.

|  |
| --- |
| Set-ADUser -Identity "johndoe" -Description "New description"  Get-ADUser -Identity "johndoe" -Properties \* |

1. Insert a screenshot of the results.

Click or tap here to enter text.

1. Disable a user account with **Disable-ADAccount**.

|  |
| --- |
| Disable-ADAccount -Identity "johndoe"  Get-ADUser -Identity "johndoe" -Properties \* |

1. Insert a screenshot of the results.

Click or tap here to enter text.

1. To enable a disabled user account, use **Enable-ADAccount**.

|  |
| --- |
| Enable-ADAccount -Identity "johndoe"  Get-ADUser -Identity "johndoe" -Properties \* |

1. Insert a screenshot of the results.

Click or tap here to enter text.

1. Reset a user's password using **Set-ADAccountPassword**.

|  |
| --- |
| Set-ADAccountPassword -Identity "johndoe" -NewPassword (ConvertTo-SecureString "NewP@ssw0rd" -AsPlainText -Force) |

1. Insert a screenshot of the results.

Click or tap here to enter text.

1. Remove a user account with **Remove-ADUser**:

|  |
| --- |
| Remove-ADUser -Identity "johndoe" -Confirm:$false |

1. Insert a screenshot of the results.

Click or tap here to enter text.

## Assignment Submission

1. Attach the program files.
2. Attach this completed document.
3. Submit in Blackboard.