# Lab-1-1: Introduction to PowerShell

# Objectives

* Gather a basic understanding of and experience with Microsoft PowerShell



# Preparation

1. Open the PowerShell ISE (To do that type ‘powershell’ in the search field on your workstation
2. Select Windows **PowerShell ISE**). You can also perform this lab in the basic PowerShell environment (Windows **PowerShell**), but you may be missing the code completion facilities of the Interactive Scripting Environment (ISE).
3. You can also pin it to your taskbar for ease of access in future.

# Exploring PowerShell

Solve the following tasks based on the **integrated help** (get-help), **lecture slides** and PowerShell **online reference**.

**Write down the commands you used to achieve the tasks.**

* Press Ctr+r. Press Ctr+r again. As you can see you are switching between windows.
* Type ls. Click the run button or F5.

ls is a bash command, but you are using powershell. Lets check out if there are more bash commands available.

**Q1.** Type alias. Find at least five bash commands that you are familiar with from the list and write those along with their aliases below:

**Ans:**

**cat = Get-Content**

**Ls = Get-Childitem**

**Md = mkdir**

**Kill = stop-process**

* Try the cmd-lets instead of the bash commands and see if they work

**Q2.** What is the use of aliases?

**Ans:**

they provide an alternative way to invoke a command, using simpler or easier to remember command

* Find help about get-childitem command let. Use -detailed, -online and -examples options (use tab for auto completion).

**Q3**. Use get-childitem to show all files on your Desktop. Type the command as answer below:

**Ans:**

PS C:\Users\leggtc1> Get-ChildItem C:\Users\leggtc1\Desktop

Directory: C:\Users\leggtc1\Desktop

Mode LastWriteTime Length Name

---- ------------- ------ ----

-a---- 21/07/2021 7:45 am 2368 Microsoft Teams.lnk

-a---- 21/07/2021 7:44 am 395 OneDrive Web.url

-a---- 21/07/2021 7:48 am 83900 Screenshot 2021-07-21 074854.jpg

-a---- 21/07/2021 7:44 am 121 Student E-mail.url

* Find help for get-process cmdlet with -online option

**Q4.** Use get-process cmdlet to find the username of the owner of powershell process (you might have to launch powershell with administrative privilege)

**Ans:**

PS C:\WINDOWS\system32> Get-Process -Name powershell\_ise

Handles NPM(K) PM(K) WS(K) CPU(s) Id SI ProcessName

------- ------ ----- ----- ------ -- -- -----------

1124 58 171600 168920 9.78 8656 1 powershell\_ise

1095 58 207900 228920 31.03 10672 1 powershell\_ise

**Q5.** Find the *username* of the owner of powershell using its process id

Ans:

PS C:\WINDOWS\system32> Get-Process -Id 10672 -IncludeUserName

Handles WS(K) CPU(s) Id UserName ProcessName

------- ----- ------ -- -------- -----------

1095 228904 31.03 10672 OPNET\leggtc1 powershell\_ise

**Q6.** Try to find the commands whose names start with ‘Write’

Construct the command in two steps:

* 1. First show all commands.
  2. Filter the results using ‘-Name write\*’.

**Ans:**

PS C:\WINDOWS\system32> Get-Command

CommandType Name Version Source

----------- ---- ------- ------

Alias Add-AppPackage 2.0.1.0 Appx

Alias Add-AppPackageVolume 2.0.1.0 Appx

Microsoft.PowerShell.Utility

Cmdlet Write-Verbose 3.1.0.0 Microsoft.PowerShell.Utility

Cmdlet Write-Warning 3.1.0.0 Microsoft.PowerShell.Utility

PS C:\WINDOWS\system32> Get-Command -Name "write\*"

CommandType Name Version Source

----------- ---- ------- ------

Alias write -> Write-Output

Alias Write-FileSystemCache 2.0.0.0 Storage

Function Write-DtcTransactionsTraceSession 1.0.0.0 MsDtc

Function Write-PrinterNfcTag 1.1 PrintManagement

Function Write-VolumeCache 2.0.0.0 Storage

Cmdlet Write-Debug 3.1.0.0 Microsoft.PowerShell.Utility

Cmdlet Write-Error 3.1.0.0 Microsoft.PowerShell.Utility

Cmdlet Write-EventLog 3.1.0.0 Microsoft.PowerShell.Management

Cmdlet Write-Host 3.1.0.0 Microsoft.PowerShell.Utility

Cmdlet Write-Information 3.1.0.0 Microsoft.PowerShell.Utility

Cmdlet Write-Output 3.1.0.0 Microsoft.PowerShell.Utility

Cmdlet Write-Progress 3.1.0.0 Microsoft.PowerShell.Utility

Cmdlet Write-Verbose 3.1.0.0 Microsoft.PowerShell.Utility

Cmdlet Write-Warning 3.1.0.0 Microsoft.PowerShell.Utility

Application write.exe 10.0.19... C:\WINDOWS\system32\write.exe

Application write.exe 10.0.19... C:\WINDOWS\write.exe

**Q7.** Create a new file on your C drive using the cmdlet new-item. Try other

ways of creating files (e.g. using *echo* or *touch*) and ensure they work.

**Ans**

PS C:\> new-item newFile.txt

Directory: C:\

Mode LastWriteTime Length Name

---- ------------- ------ ----

-a---- 21/07/2021 9:43 am 0 newFile.txt

New-item -path $profile -itemtype “file” -Force

**Q8**. Everything is an object. Play around with this.

* 1. Show the current date.
  2. Pipe it into ‘| format-list’ to show all properties.
  3. Explore the properties using get-member.

**Ans**:

PS C:\> Get-Date

Wednesday, 21 July 2021 9:46:02 am

PS C:\> Get-Date | format-list

DisplayHint : DateTime

Date : 21/07/2021 12:00:00 am

Day : 21

DayOfWeek : Wednesday

DayOfYear : 202

Hour : 9

Kind : Local

Millisecond : 791

Minute : 45

Month : 7

Second : 31

Ticks : 637624575317917902

TimeOfDay : 09:45:31.7917902

Year : 2021

DateTime : Wednesday, 21 July 2021 9:45:31 am

PS C:\> Get-Date | Get-Member

TypeName: System.DateTime

Name MemberType Definition

---- ---------- ----------

Add Method datetime Add(timespan value)

AddDays Method datetime AddDays(double value)

AddHours Method datetime AddHours(double value)

AddMilliseconds Method datetime AddMilliseconds(double value)

AddMinutes Method datetime AddMinutes(double value)

AddMonths Method datetime AddMonths(int months)

AddSeconds Method datetime AddSeconds(double value)

AddTicks Method datetime AddTicks(long value)

AddYears Method datetime AddYears(int value)

**Q9.** Show all folders and subfolder on the your Desktop. Do it in two steps. Use

the parameter ‘–Recurse’ to recursively show the files and folders.

[If you are receiving errors, identify why you got those. Which statement part is causing those? Adding the command ‘–ErrorAction silentlyContinue’ will solve this.]

**Ans:**

PS C:\Users\Ant> Get-ChildItem -Path "C:\Users\Ant\Desktop" -Recurse

**Q10.** Count the files and folders on C: drive. (Note: Extend the previous command.)

**Ans:**

PS C:\Users\Ant> (Get-Item 'C:\Users\Ant\Desktop\\*').Length

23

**Q11.** Write the names of all files in C: to the file “files.log”. (Look at the cmdlet

*out-file*.) You can also redirect the output to the clipboard by piping it to ‘clip’.

**Ans:**

PS C:\Users\Ant> Get-ChildItem -Path "C:\Users\Ant\Desktop" -Recurse | Select BaseName | Out-File -FilePath "C:\Users\Ant\Desktop\out.txt"

**Q12** What does the function ‘ft’ stand for?

Ans:

**Q13.** Show the top 15 processes that use most processor time.

Do it in three steps and pipeline the results:

* 1. Get all processes.
  2. Sort them based on CPU usage (highest usage first). Check the help for the ‘sort’ command
  3. Select the first 15.

**Ans:**

**Q14.** Determine the total used CPU time (Hint: have a look at *get-process*).

Ans: