Module 3:
Working with the Windows PowerShell pipeline

Lab A: Using the pipeline

Exercise 1: Selecting, sorting, and displaying data

Task 1: Display the current day of the year

1. On **LON-CL1**, click **Start** and then type **powersh**.

2. In the search results, right-click **Windows PowerShell**, and then click **Run as administrator**.

3. In the **Administrator: Windows PowerShell** window, type the following command, and then press Enter: ***help *date****

Note: *Notice the* ***Get-Date*** *command.*

4. In the console, type the following command, and then press Enter:
Get-Date | Get-Member

Note: *Notice the* ***DayOfYear*** *property.*

5. In the console, type the following command, and then press Enter:
Get-Date | Select-Object –Property DayOfYear

6. In the console, type the following command, and then press Enter:
Get-Date | Select-Object -Property DayOfYear | fl

Task 2: Display information about installed hotfixes

1. In the console, type the following command, and then press Enter:
Get-Command *hotfix*

Note: *Notice the* ***Get-Hotfix*** *command.*

2. In the console, type the following command, and then press Enter:
Get-Hotfix | Get-Member

Note: *The properties of the* ***Hotfix*** *object display. If needed, run* ***Get-Hotfix*** *to see some of the values that typically appear in those properties.*

3. In the console, type the following command, and then press Enter:
Get-Hotfix | Select-Object –Property HotFixID,InstalledOn,InstalledBy

4. In the console, type the following command, and then press Enter:
Get-Hotfix | Select-Object –Property HotFixID,@{n='HotFixAge';e={(New-TimeSpan –Start $PSItem.InstalledOn).Days}},InstalledBy

Task 3: Display a list of available scopes from the DHCP server

1. In the console, type the following command, and then press Enter:
help *scope*

Note: *Notice the* ***Get-DHCPServerv4Scope*** *command.*

2. In the console, type the following command, and then press Enter:
Help Get-DHCPServerv4Scope –ShowWindow

Note: *Notice the available parameters.*

3. In the console, type the following command, and then press Enter:
Get-DHCPServerv4Scope –ComputerName LON-DC1

4. In the console, type the following command, and then press Enter:
Get- DHCPServerv4Scope –ComputerName LON-DC1 | Select-Object –Property ScopeId,SubnetMask,Name | fl

Task 4: Display a sorted list of enabled Windows Firewall rules

1. In the console, type the following command, and then press Enter: ***help *rule****

Note: *Notice the* ***Get-NetFirewallRule*** *command.*

2. In the console, type the following command, and then press Enter:
Get-NetFirewallRule

3. In the console, type the following command, and then press Enter:
Help Get-NetFirewallRule –ShowWindow

4. In the console, type the following command, and then press Enter:
Get-NetFirewallRule –Enabled True

5. In the console, type the following command, and then press Enter:
Get-NetFirewallRule –Enabled True | Format-Table -wrap

6. In the console, type the following command, and then press Enter:
Get-NetFirewallRule –Enabled True | Select-Object –Property DisplayName,Profile,Direction,Action | Sort-Object –Property Profile, DisplayName | ft -GroupBy Profile

Task 5: Display a sorted list of network neighbors

1. In the console, type the following command, and then press Enter:
help *neighbor*

Note: *Notice the* ***Get-NetNeighbor*** *command.*

2. In the console, type the following command, and then press Enter:
help Get-NetNeighbor –ShowWindow

3. In the console, type the following command, and then press Enter:
Get-NetNeighbor

4. In the console, type the following command, and then press Enter:
Get-NetNeighbor | Sort-Object –Property State

5. In the console, type the following command, and then press Enter:
Get-NetNeighbor | Sort-Object –Property State | Select-Object –Property IPAddress,State | Format-Wide -GroupBy State -AutoSize

Task 6: Display information from the DNS name resolution cache

1. In the console, type the following command, and then press Enter:
Test-NetConnection LON-DC1

2. In the console, type the following command, and then press Enter:
Test-NetConnection LON-CL1

3. In the console, type the following command, and then press Enter:
help *cache*

Note: *Notice the* ***Get-DnsClientCache*** *command.*

4. In the console, type the following command, and then press Enter:
Get-DnsClientCache

5. In the console, type the following command, and then press Enter:
Get-DnsClientCache | Select Name,Type,TimeToLive | Sort Name | Format-List

Note: *Notice that the* ***Type*** *data does not return what you might expect—for example, A and CNAME. Instead, it returns raw numerical data. Each number maps directly to a record type, and you can filter for those types when you know the map: 1= A, 5 = CNAME, and so on. Later in this module, you will learn how to add more filters to determine the numbers and their corresponding record types. You will notice a similar situation for other returned data, such as* ***Status*** *data.*

6. Close all open windows.

Results*: After completing this exercise, you should have produced several custom reports that contain management information from your environment.*

Task 7: Prepare for the next lab

• At the end of this lab, keep the virtual machines running as they are needed for the next lab.

Lab B: Filtering objects

Exercise 1: Filtering objects

Task 1: Display a list of all the users in the Users container of Active Directory

1. On **LON-CL1**, click **Start** and then type **powersh**.

2. In the search results, right-click **Windows PowerShell**, and then click **Run as administrator**.

3. In the **Administrator: Windows PowerShell** window, type the following command, and then press Enter: ***help *user****

Note: *Notice the* ***Get-ADUser*** *command.*

4. In the console, type the following command, and then press Enter:
Get-Help Get-ADUser –ShowWindow

Note: *Notice that the -Filter parameter is mandatory. Review the examples for the command.*

5. In the console, type the following command, and then press Enter:
Get-ADUser –Filter * | ft

6. In the console, type the following command, and then press Enter:
Get-ADUser –Filter * -SearchBase "cn=Users,dc=Adatum,dc=com" | ft

Task 2: Create a report showing the Security event log entries that have the event ID 4624

1. In the console, type the following command, and then press Enter:
Get-EventLog -LogName Security | Where EventID -eq 4624 | Measure-Object | fw

2. In the console, type the following command, and then press Enter:
Get-EventLog -LogName Security | Where EventID -eq 4624 | Select TimeWritten,EventID,Message

3. In the console, type the following command, and then press Enter:
Get-EventLog -LogName Security | Where EventID -eq 4624 | Select TimeWritten,EventID,Message -Last 10 | fl

Task 3: Display a list of the encryption certificates installed on the computer

1. In the console, type the following command, and then press Enter:
Get-ChildItem -Path CERT: -Recurse

2. To display the members of the objects, type the following command, and then press Enter: ***Get-ChildItem -Path CERT: -Recurse | Get-Member***

3. In the console, type one of the following commands, and then press Enter:
Get-ChildItem -Path CERT: -Recurse | Where HasPrivateKey -eq $False | Select-Object -Property FriendlyName,Issuer | fl
or:
Get-ChildItem -Path CERT: -Recurse | Where { $PSItem.HasPrivateKey -eq $False } | Select-Object –Property FriendlyName,Issuer | fl

4. In the console, type the following command, and then press Enter:
Get-ChildItem -Path CERT: -Recurse | Where { $PSItem.HasPrivateKey -eq $False -and $PSItem.NotAfter -gt (Get-Date) –and $PSItem.NotBefore -lt (Get-Date) } | Select-Object -Property NotBefore,NotAfter, FriendlyName,Issuer | ft -wrap

Task 4: Create a report that shows the disk volumes that are running low on space

1. In the console, type the following command, and then press Enter:
Get-Volume

Note: *If you did not know the command name, you could have run* ***Help *volume**** *to discover it.*

2. In the console, type the following command, and then press Enter:
Get-Volume | Get-Member

Note: *Notice the* ***SizeRemaining*** *property.*

3. In the console, type the following command, and then press Enter:
Get-Volume | Where-Object { $PSItem.SizeRemaining -gt 0 } | fl

4. In the console, type the following command, and then press Enter:
Get-Volume | Where-Object { $PSItem.SizeRemaining -gt 0 -and $PSItem.SizeRemaining / $PSItem.Size -lt .99 }| Select-Object DriveLetter, @{n='Size';e={'{0:N2}' –f ($PSItem.Size/1MB)}}

5. In the console, type the following command, and then press Enter:
Get-Volume | Where-Object { $PSItem.SizeRemaining -gt 0 -and $PSItem.SizeRemaining / $PSItem.Size -lt .1 }

Note: *This command might not produce any output on your lab computer if the computer has more than 10 percent free space on each of its volumes.*

Task 5: Create a report that displays specified Control Panel items

1. In the console, type the following command, and then press Enter:
help *control*

Note: *Notice the* ***Get-ControlPanelItem*** *command.*

2. In the console, type the following command, and then press Enter:
Get-ControlPanelItem

3. In the console, type the following command, and then press Enter:
Get-ControlPanelItem –Category 'System and Security' | Sort Name

Note: *Notice that you do not have to use* ***Where-Object****.*

4. In the console, type the following command, and then press Enter:

Get-ControlPanelItem -Category 'System and Security' | Where-Object -FilterScript {-not ($PSItem.Category -notlike '*System and Security*')} | Sort Name

***Results**: After completing this exercise, you should have used filtering to produce lists of management information that include only specified data and elements.*

Task 6: Prepare for the next lab

• At the end of this lab, keep the virtual machines running as they are needed for the next lab.

Lab C: Enumerating objects

Exercise 1: Enumerating objects

Task 1: Display a list of files on the E: drive of your computer

1. On **LON-CL1**, click **Start** and then type **powersh**.

2. In the search results, right-click **Windows PowerShell**, and then click **Run as administrator**.

3. In the **Administrator: Windows PowerShell** window, type the following command, and then press Enter: ***Get-ChildItem -Path E: -Recurse***

4. In the console, type the following command, and then press Enter:
Get-ChildItem -Path E: -Recurse | Get-Member

***Note:** Notice the **GetFiles** method in the list under **TypeName: System.IO.DirectoryInfo**.*

***Note:** You will learn why there are two lists of members for Get-ChildItem in Module 5, "Using PSProviders and PSDrives."*

5. In the console, type the following command, and then press Enter:
Get-ChildItem -Path E: -Recurse | ForEach GetFiles

Task 2: Use enumeration to produce 100 random numbers

1. In the console, type the following command, and then press Enter:
help *random*

Note: *Notice the* ***Get-Random*** *command.*

2. In the console, type the following command, and then press Enter:
help Get-Random –ShowWindow

Note: *Notice the -SetSeed parameter.*

3. In the console, type the following command, and then press Enter: ***1..100***

4. In the console, type the following command, and then press Enter:
1..100 | ForEach { Get-Random –SetSeed $PSItem }

Task 3: Run a method of a Windows Management Instrumentation (WMI) object

1. Close all applications other than the **Windows PowerShell** console.

2. In the console, type the following command, and then press Enter:
Get-WmiObject –Class Win32_OperatingSystem -EnableAllPrivileges

3. In the console, type the following command, and then press Enter:
Get-WmiObject –Class Win32_OperatingSystem -EnableAllPrivileges | Get-Member

Note: *Notice the* ***Reboot*** *method.*

Note: *The following command will reboot the machine you run it on.*

4. In the console, type the following command, and then press Enter:
Get-WmiObject –Class Win32_OperatingSystem -EnableAllPrivileges | ForEach Reboot

Results*: After completing this exercise, you should have written commands that manipulate multiple objects in the pipeline.*

Task 4: Prepare for the next lab

• At the end of this lab, keep the virtual machines running as they are needed for the next lab.

Lab D: Sending output to a file

Exercise 1: Converting objects

Task 1: Update Active Directory user information

Note: *In this lab, long commands typically display on several lines. This helps to prevent unintended line breaks in the middle of commands. However, when you type these commands, you should type them as a single line. That line might wrap on your screen into multiple lines, but the command will still work. Press Enter only after typing the entire command.*

1. Sign in to the **LON-CL1** as **Adatum\Administrator** with the password of **Pa55w.rd.**

2. On **LON-CL1**, click **Start** and then type **powersh**.

3. In the search results, right-click **Windows PowerShell**, and then click **Run as administrator**.

4. In the **Administrator: Windows PowerShell** window, type the following command, and then press Enter:
Get-ADUser -Filter * -Properties Department,City | Where {$PSItem.Department -eq 'IT' -and $PSItem.City -eq 'London'} | Select-Object -Property Name,Department,City| SortName

5. In the **Administrator: Windows PowerShell** window, type the following command, and then press Enter:
Get-ADUser -Filter * -Properties Department,City | Where {$PSItem.Department -eq 'IT' -and $PSItem.City -eq 'London'} | Set-ADUser -Office 'LON-A/1000'

6. In the **Administrator: Windows PowerShell** window, type the following command, and then press Enter:
Get-ADUser -Filter * -Properties Department,City,Office | Where {$PSItem.Department -eq 'IT' -and $PSItem.City -eq 'London'} | Select-Object –Property Name,Department,City,Office | Sort Name

Task 2: Produce an HTML report listing the Active Directory users in the IT department

1. In the console, type the following command, and then press Enter:
help ConvertTo-Html –ShowWindow

2. In the console, type the following command, and then press Enter:
Get-ADUser -Filter * -Properties Department,City,Office | Where {$PSItem.Department -eq 'IT' -and $PSItem.City -eq 'London'} | Sort Name | Select-Object -Property Name,Department,City,Office | ConvertTo-Html –Property Name,Department,City -PreContent Users | Out-File E:\UserReport.html

3. To view the HTML file, type the following command, and then press Enter:
Invoke-Expression E:\UserReport.html

4. In the console, type the following command, and then press Enter:
Get-ADUser -Filter * -Properties Department,City,Office | Where {$PSItem.Department -eq 'IT' -and $PSItem.City -eq 'London'} | Sort Name | Select-Object -Property Name,Department,City,Office | Export-Clixml E:\UserReport.xml

5. In Internet Explorer, in the address bar, type ***E:\UserReport.xml***, and then press Enter.

6. In the console, type the following command, and then press Enter:
Get-ADUser -Filter * -Properties Department,City,Office | Where {$PSItem.Department -eq 'IT' -and $PSItem.City -eq 'London'} | Sort Name | Select-Object -Property Name,Department,City,Office | Export-Csv E:\UserReport.csv

7. In File Explorer, go to **E:**, right-click **UserReport.csv**, click **Open with**, and then click **Notepad**.

8. In File Explorer, go to **E:**, double-click **UserReport.csv**.

Results*: After completing this exercise, you should have converted Active Directory user objects to different data formats.*

Task 3: Prepare for the next module

When you have finished the lab, revert the virtual machines to their initial state. To do this, perform the following steps:

1. On the host computer, start **Hyper-V Manager**.
2. In the **Virtual Machines** list, right-click **10961C-LON-DC1**, and then click **Revert**.
3. In the **Revert Virtual Machine** dialog box, click **Revert**.
4. Repeat steps 2 and 3 for **10961C-LON-CL1**.