WELCOME TO MODULE 7

WINDOWS SCRIPTING & COMMAND LINE

HERE BELOW ARE ALL SECTION THAT WE ARE GOING TO COVER IN THIS MODULE.

- Section I- WINDOWS BATCH SCRIPTING
- Section II- FIRST BATCH SCRIPT

- Section III- INTRODUCTION TO POWERSHELL & ITS IMPORTANCE
- Section IV- SCRIPT TO AUTOMATE SIMPLE TASKS
- Section VI- WINDOWS POWERSHELL
- Section VII- POWERSHELL COMMAND LINE
- Section VIII- WINDOWS POWERSHELL ISE
- Section IX- WINDOWS MANAGEMENT INSTRUMENT (WMIC)
- Section X- DIFFERENCE BETWEEN DOS & POWERSHELL

SECTION I- WINDOWS BATCH SCRIPTING

- A batch file is a kind of script file in Microsoft Windows. It consists of a series of commands to be executed by the command-line interpreter, stored in a plain text file
- All the commands run on DOS windows can be combined in a file and executed as a batch file
- The filename extension .bat is used in DOS and Windows which tells the OS that it is executable

For example

Open up a notepad and write

```
dir
timeout 5
```

Save as .bat file and execute. This will list the files in a directory and wait for 5 seconds before exiting

SECTION II- FIRST BATCH SCRIPT

DOS batch script

- Open up a notepad and write

```
echo Hello World
timeout 5
```

- Save as .bat file and execute. This will print "Hello World" on the DOS windows and wait for 5 seconds before exiting

Batch script for pop-up windows

- Open up a notepad and write

```
msg * "Hello World"
```

SCRIPT TO AUTOMATE SIMPLE TASKS

- Start a calculator
- Open File explorer
- Open paint application
- Task Manager

SECTION III- INTRODUCTION TO POWERSHELL & ITS IMPORTANCE

- PowerShell is a task automation and configuration management framework from Microsoft, consisting of a command-line shell and associated scripting language
- PowerShell includes its own extensive, console-based help (similar to man pages in Unix or Linux shells) accessible via the **`Get-Help`** cmdlet (command let)
- First run **`Update-Help`** before getting any help on a command

  e.g. **`Get-Help <keyword> remote`**

  `Get-Help Get-Process`

- To get a list of all commands available

  `Get-Command.`

WHY POWERSHELL

Why PowerShell

Consistency.

A scripted solution will run the exact same script every time

No risk of typos, forgetting to complete the task, or doing the task incorrectly

Audit trail.

There are many tasks where having an audit trail would be helpful, perhaps including what task was performed, important results, errors that occurred, when the task ran, who ran it, and so forth.

Change of pace.

- From Scripting you can achieve a task significantly faster than GUI

PowerShell is easy to adopt, learn, and use because it does not require a background in programming.

Purpose of PowerShell

- Improved Management
- Improved Automation
- Manage real-time
- Manage large scale environments

Get-Help

The **Get-Help** cmdlet displays information about Windows PowerShell concepts and commands, including cmdlets, functions, CIM commands, workflows, providers, aliases and scripts.

Example:

Get-Command -Name "***service***"

Get-Help Get-Service	# Get Help available for any PowerShell command
Get-Help Get-Service **–Full**	# To access full help with examples
Get-Help Get-Service **–online**	# Search for online help
Get-Help Get-Service **-ShowWindow**	#Special window for navigation through Help

SECTION VII- POWERSHELL COMMAND LINE

- `Get-ChildItem -Path C:\` (Listing All the Files and Folders Within a Folder)
- `New-Item -Path 'C:\shows\Seinfeld' -ItemType Directory`
- `New-Item -Path 'C:\shows\Seinfeld\jerry.txt' -ItemType File`
- `Copy-Item -Path C:\shows\Seinfeld\jerry.txt -Destination C:\jerry.bak`
- `Remove-Item -Path C:\shows\Seinfeld\jerry.bak`
- `Stop-computer, Restart-Computer`
- `Get-Process, Get-Service, Get-PSDrive`

- PowerShell commands can be automated

Open up a notepad and write

```
echo "Hello World"
sleep 5
```

- Save as .ps1 file and execute. This will print "Hello World" on the PowerShell windows and wait for 5 seconds before exiting

SECTION VIII- WINDOWS POWERSHELL ISE

- PowerShell is developed for configuration management and automation of tasks. ... PowerShell ISE is expanded as PowerShell Integrated scripting environment which is a graphical user interface that enables the user to execute commands, create and modify the scripts directly on the GUI instead of typing as a command.

- On top of the standard command-line shell, you can also find the Windows PowerShell ISE. ISE stands for Integrated Scripting Environment, and it is a graphical user interface that allows you to easily create different scripts without having to type all the commands in the command line

- In Windows PowerShell ISE, you can run commands and write, test, and debug scripts in a single Windows-based graphic user interface.

- The interface provides multiline editing, tab completion, syntax coloring, selective execution, context-sensitive help, and support for right-to-left languages.

Section IX- WINDOWS MANAGEMENT INSTRUMENT (WMIC)

- Windows Management Instrumentation (WMI) is the infrastructure for management data and operations on Windows–based operating systems. You can write WMI scripts or applications to automate administrative tasks on remote computers but WMI also supplies management data to other parts of the operating system and products

- To start wmi command prompt
  - Run **`wmic`** from DOS
  - enter **`/?`** for Help
  - Simple commands like: **`memorychip`**, **`ntdomain`**, **`os`** etc.
  - A commands cheat sheet has been attached in handouts section

- To access wmi console
  - **`wmimgmt.msc`**

- WMI(Windows management Instrumentation) is Microsoft's implementation of the Web-Based Enterprise Management (WBEM) and Common Information Model (CIM) standards from the Distributed Management Task Force (DMTF).

- WMI allows scripting languages (such as VBScript or Windows PowerShell) to manage Microsoft Windows personal computers and servers, both locally and remotely. WMI comes preinstalled in Windows 2000 and in newer Microsoft OSes.

WMI Purpose

- WMI provides users with information about the status of local or remote computer systems.
- It also supports such actions as the configuration of security settings, setting and changing system properties, setting and changing permissions for authorized users and user groups, assigning and changing drive labels, scheduling processes to run at specific times, backing up the object repository, and enabling or disabling error logging.

Difference Between DOS and PowerShell

- PowerShell is actually very different from the Command Prompt. It uses different commands, known as cmdlets in PowerShell. Many system administration tasks — from managing the registry to WMI (Windows Management Instrumentation) — are exposed via PowerShell cmdlets, while they aren't accessible from the Command Prompt

- The first version of PowerShell, which is based on the .NET framework, was released way back in 2006 and is much more advanced than the Command Prompt. PowerShell has many different advanced features like command piping, task automation, remote execution, etc.

- All this power of PowerShell comes at a cost; that is the learning curve. If you don’t mind the steep learning curve, then do give PowerShell a try