

PowerShell & PSADT

Date: 09-August-2025

Batch: WiproNGA_DWS_B5_25VID2550

Candidate Name: Sooraj B

User ID: 34753

1. PSADT (PowerShell App Deployment Toolkit) Overview

The PowerShell App Deployment Toolkit simplifies software deployments in enterprise environments by offering a structured and flexible scripting framework. It's widely used with SCCM and Intune.

Key Components:

- **Files** → Application installers
- **SupportFiles** → Dependencies/configs
- **Deploy-Application.ps1** → Main script for install, uninstall, and updates
- **Logging** → Stored in C:\Windows\Logs\Software (viewable in CMTrace)

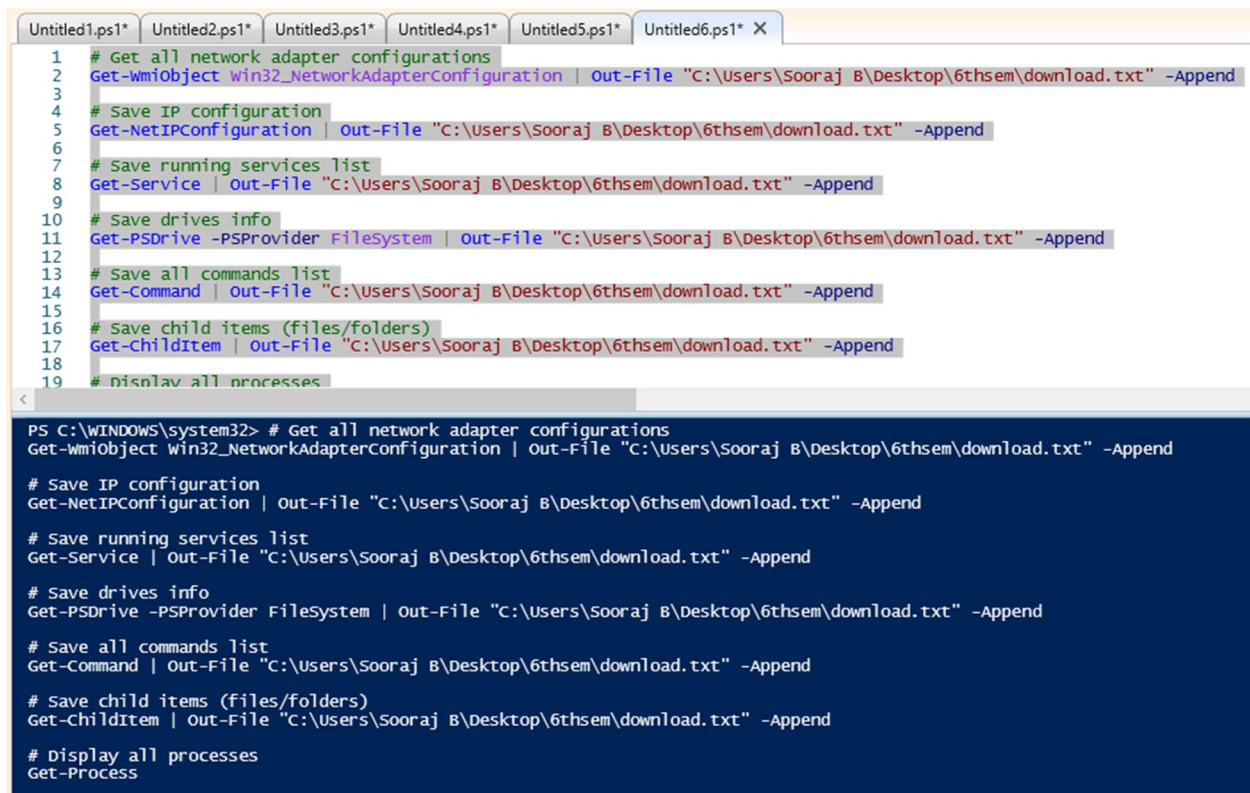
Example Install Code:

powershell

CopyEdit

Execute-MSI -Action Install -Path "AppInstaller.msi"

2. PowerShell Scripting Constructs

The image shows a screenshot of a PowerShell script editor with six tabs labeled 'Untitled1.ps1*' through 'Untitled6.ps1* X'. The script in the active tab contains 19 lines of PowerShell commands. Lines 1-19 are: 1: # Get all network adapter configurations, 2: Get-WmiObject win32_NetworkAdapterConfiguration | Out-File "C:\Users\Sooraj B\Desktop\6thsem\download.txt" -Append, 3: (blank), 4: # Save IP configuration, 5: Get-NetIPConfiguration | Out-File "C:\Users\Sooraj B\Desktop\6thsem\download.txt" -Append, 6: (blank), 7: # Save running services list, 8: Get-Service | Out-File "C:\Users\Sooraj B\Desktop\6thsem\download.txt" -Append, 9: (blank), 10: # Save drives info, 11: Get-PSDrive -PSProvider FileSystem | Out-File "C:\Users\Sooraj B\Desktop\6thsem\download.txt" -Append, 12: (blank), 13: # Save all commands list, 14: Get-Command | Out-File "C:\Users\Sooraj B\Desktop\6thsem\download.txt" -Append, 15: (blank), 16: # Save child items (files/folders), 17: Get-Childitem | Out-File "C:\Users\Sooraj B\Desktop\6thsem\download.txt" -Append, 18: (blank), 19: # Display all processes. Below the editor, a terminal window shows the execution of the script, displaying the same commands as the script editor, with the first command being 'PS C:\WINDOWS\system32> # Get all network adapter configurations' and the subsequent commands being the individual PowerShell commands from the script.

PowerShell scripting constructs are the core of automation.

Example Variables and Arrays:

powershell

CopyEdit

```
$name = "John" $age = 30 $fruits = "apple","banana" $fruits += "kiwi"
```

Example Condition:

powershell

CopyEdit

```
if ($age -ge 18) { Write-Host "Adult" } else { Write-Host "Minor" }
```

3. WMI & CIM

Used to query system hardware and OS info.

```

22 # Prompt for user name and greet
23 $name = Read-Host "Please enter name"
24 Write-Output "Hello, let's welcome $name"
25
26 # Example of formatted string
27 $name = "Sooraj B"
28 $age = 26
29 "My name is {1} and I am {0} years old" -f $name, $age
30
31 # Show top 5 processes by CPU
32 Get-Process | Sort-Object CPU -Descending | Select-Object -First 5 | Select-Object Name, CPU
33
34 # Working with arrays
35 $fruits = "banana","apple"
36 $fruits[0]
37 $fruits += "kiwi"
38 $fruits[2]
39
40 # IF condition example

```

```

# Prompt for user name and greet
$name = Read-Host "Please enter name"
Write-Output "Hello, let's welcome $name"

# Example of formatted string
$name = "Sooraj B"
$age = 26
"My name is {1} and I am {0} years old" -f $name, $age

# Show top 5 processes by CPU
Get-Process | Sort-Object CPU -Descending | Select-Object -First 5 | Select-Object Name, CPU

# working with arrays
$fruits = "banana","apple"
$fruits[0]
$fruits += "kiwi"
$fruits[2]

# IF condition example
$age = 20
if ($age -ge 22) {

```

Examples:

powershell

CopyEdit

```

Get-CimInstance Win32_OperatingSystem | Out-File "C:\Users\Sooraj
B\Desktop\6thsem\download.txt" -Append Get-CimInstance Win32_Processor | Out-File
"C:\Users\Sooraj B\Desktop\6thsem\download.txt" -Append

```

4. Modularization in PowerShell

Break large scripts into modules for reusability.

Example:

```
41 $age = 20
42 if ($age -ge 22) {
43     Write-Host "Age is greater"
44 }
45
46 # IF-ELSEIF-ELSE example
47 $age = 25
48 if ($age -ge 18) {
49     Write-Host "Adult"
50 } elseif ($age -ge 13) {
51     Write-Host "Teenager"
52 } else {
53     Write-Host "Child"
54 }
55
56 # Read content from file and greet
57 $data = Get-Content -Path "C:\Users\Sooraj B\Desktop\6thsem\download.txt"
58 Write-Output "Hello, let's welcome $data"
59
```

```
<
    Write-Host "Age is greater"
}

# IF-ELSEIF-ELSE example
$age = 25
if ($age -ge 18) {
    Write-Host "Adult"
} elseif ($age -ge 13) {
    Write-Host "Teenager"
} else {
    Write-Host "Child"
}

# Read content from file and greet
$data = Get-Content -Path "C:\Users\Sooraj B\Desktop\6thsem\download.txt"
Write-Output "Hello, let's welcome $data"

# Get service members
Get-Service | Get-Member | Out-File "C:\Users\Sooraj B\Desktop\6thsem\download.txt" -Append

# Get date members
```

powershell

CopyEdit

```
function Get-SystemInfo { Get-CimInstance Win32_OperatingSystem } Export-ModuleMember
-Function Get-SystemInfo
```

5. Input, Output & Formatting

Example Input & Output:

```

133 Get-Process -Name "brave" | Stop-Process
134
135 # Show service status
136 Get-Service | ForEach-Object Status | Out-File "C:\Users\Sooraj B\Desktop\6thsem\download.txt" -Append
137
138 # Show service display names
139 Get-Service | ForEach-Object DisplayName | Out-File "C:\Users\Sooraj B\Desktop\6thsem\download.txt" -Append
140
141 # Show event logs
142 Get-EventLog -List | Out-File "C:\Users\Sooraj B\Desktop\6thsem\download.txt" -Append
143
144 # Show only system log
145 Get-EventLog -List | Where-Object Log -eq "System" | Out-File "C:\Users\Sooraj B\Desktop\6thsem\download.txt" -Append
146
147 # Convert process list to HTML
148 Get-Process | ConvertTo-Html | Out-File "C:\Users\Sooraj B\Desktop\6thsem\download.txt" -Append
149

```

```

Get-EventLog -List | Where-Object Log -eq "System" | Out-File "C:\Users\Sooraj B\Desktop\6thsem\download.txt" -Append
# Convert process list to HTML
Get-Process | ConvertTo-Html | Out-File "C:\Users\Sooraj B\Desktop\6thsem\download.txt" -Append

```

Handles	NPM(K)	PM(K)	WS(K)	CPU(s)	Id	SI	ProcessName
191	11	4380	5552	7.86	7632	0	AggregatorHost
235	15	22448	456	7.47	20668	1	ai
153	10	2408	148	1.28	3264	0	amdfendrsr
300	17	16380	4508	4.53	9248	1	AppActions
302	20	4784	1680	118.09	6456	1	AppleMobileDeviceProcess
454	26	13028	16408	18.69	14120	1	ApplicationFrameHost
682	17	4772	3120	220.92	3484	1	atieclxx
200	10	1600	552	0.33	3276	0	atiesrxx
329	28	17860	404	0.41	11780	1	backgroundTaskHost
331	28	17804	416	0.42	15916	1	backgroundTaskHost
329	28	17988	420	0.98	17600	1	backgroundTaskHost
331	28	17508	408	0.77	24288	1	backgroundTaskHost
330	28	17504	404	0.50	25212	1	backgroundTaskHost

powershell

CopyEdit

```
$name = Read-Host "Enter your name" Write-Output "Hello, $name" | Out-File "C:\Users\Sooraj B\Desktop\6thsem\download.txt" -Append
```

Formatting Example:

powershell

CopyEdit

```
Get-Service | Format-Table Name, Status | Out-File "C:\Users\Sooraj B\Desktop\6thsem\download.txt" -Append
```

6. Combined

```

1  # Script: SystemInfo.ps1
2  # Author: Sooraj B
3  # Description: Displays basic system information
4
5  Write-Host "Sooraj B"
6  Write-Host "-----"
7  |
8  $COMPUTERNAME = $env:COMPUTERNAME
9  $os = Get-CimInstance win32_operatingsystem
10 $cpu = Get-CimInstance win32_Processor
11
12 Write-Host "Computer Name: $COMPUTERNAME"
13 Write-Host "Operating System: $($os.Caption)"
14 Write-Host "CPU: $($cpu.Name)"
15

```

```

PS C:\WINDOWS\system32> # Script: SystemInfo.ps1
# Author: Sooraj B
# Description: Displays basic system information

Write-Host "Sooraj B"
Write-Host "-----"

$COMPUTERNAME = $env:COMPUTERNAME
$os = Get-CimInstance win32_operatingsystem
$cpu = Get-CimInstance win32_Processor

Write-Host "Computer Name: $COMPUTERNAME"
Write-Host "Operating System: $($os.Caption)"
Write-Host "CPU: $($cpu.Name)"

Sooraj B
-----
Computer Name: NEC0822051
Operating System: Microsoft windows 11 Pro
CPU: AMD Athlon Silver 3050U with Radeon Graphics

```

powershell

CopyEdit

```

Get-WmiObject Win32_NetworkAdapterConfiguration | Out-File "C:\Users\Sooraj
B\Desktop\6thsem\download.txt" -Append Get-NetIPConfiguration | Out-File "C:\Users\Sooraj
B\Desktop\6thsem\download.txt" -Append Get-Service | Out-File "C:\Users\Sooraj
B\Desktop\6thsem\download.txt" -Append Get-PSDrive -PSProvider FileSystem | Out-File
"C:\Users\Sooraj B\Desktop\6thsem\download.txt" -Append Get-Command | Out-File
"C:\Users\Sooraj B\Desktop\6thsem\download.txt" -Append Get-ChildItem | Out-File
"C:\Users\Sooraj B\Desktop\6thsem\download.txt" -Append Get-Process | Out-File
"C:\Users\Sooraj B\Desktop\6thsem\download.txt" -Append Get-Hotfix | Out-File
"C:\Users\Sooraj B\Desktop\6thsem\download.txt" -Append Get-NetFirewallRule | Out-File
"C:\Users\Sooraj B\Desktop\6thsem\download.txt" -Append

```



```

1 Clear-Host
2 Get-CimInstance Win32_OperatingSystem
3 Get-Command Get-CimInstance
4 Get-Help Get-CimInstance
5 Import-Module BitsTransfer
6 Get-command Install-Module, Install-PSResource
7 Get-Module
8 Get-Module -ListAvailable
9 #Get-Command -Module <module-name>
10 Get-command -Module BitsTransfer
11 Remove-Module BitsTransfer
12 Get-command Get-Date, Get-Help -All | Select-object -Property Name, CommandType, Module ,PSSnapIn
13 function Get-Date {
14     Microsoft.PowerShell.Utility\Get-Date -Format "yyyy-MM-dd HH:mm:ss"
15 }
16
17 # Call your custom function
18 Get-Date
19
20 # Remove your custom function

```

```

SystemDirectory      Organization BuildNumber RegisteredUser SerialNumber      Version
-----
C:\WINDOWS\system32  26100      Sooraj B      00331-10000-00001-AA485 10.0.26100

Verb                  : Get
Noun                  : CimInstance
HelpFile              : C:\WINDOWS\system32\WindowsPowerShell\v1.0\Modules\CimCmdlets\en-US\Microsoft.Management.Infrastructure.CimCmdlets.dll-help.xml
PSSnapIn              :
Version              : 1.0.0.0
ImplementingType      : Microsoft.Management.Infrastructure.CimCmdlets.GetCimInstanceCommand
Definition            :
    Get-CimInstance [-ClassName] <string> [-ComputerName <string[]>] [-KeyOnly] [-Namespace <string>]
    [-OperationTimeoutSec <uint32>] [-QueryDialect <string>] [-Shallow] [-Filter <string>] [-Property
    <string[]>] [<CommonParameters>]

    Get-CimInstance -CimSession <CimSession[]> -ResourceUri <uri> [-KeyOnly] [-Namespace <string>]
    [-OperationTimeoutSec <uint32>] [-Shallow] [-Filter <string>] [-Property <string[]>]
    [<CommonParameters>]

    Get-CimInstance -CimSession <CimSession[]> -Query <string> [-ResourceUri <uri>] [-Namespace <string>]

```

Final Thoughts

This day covered:

- Using **PSADT** for real deployments
- Writing **structured scripts**
- Querying system data with **WMI/CIM**
- Organizing code via **modules**
- Handling **input/output & formatting**
- Building **reusable code** for automation