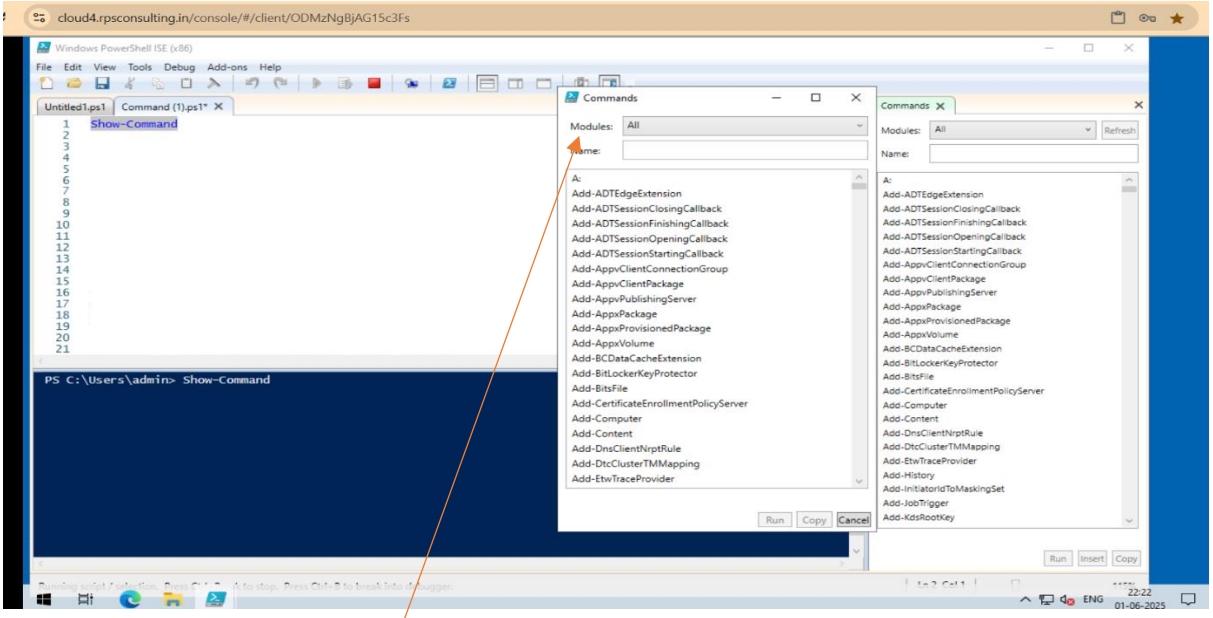
All Commands :

- **Show-Command**

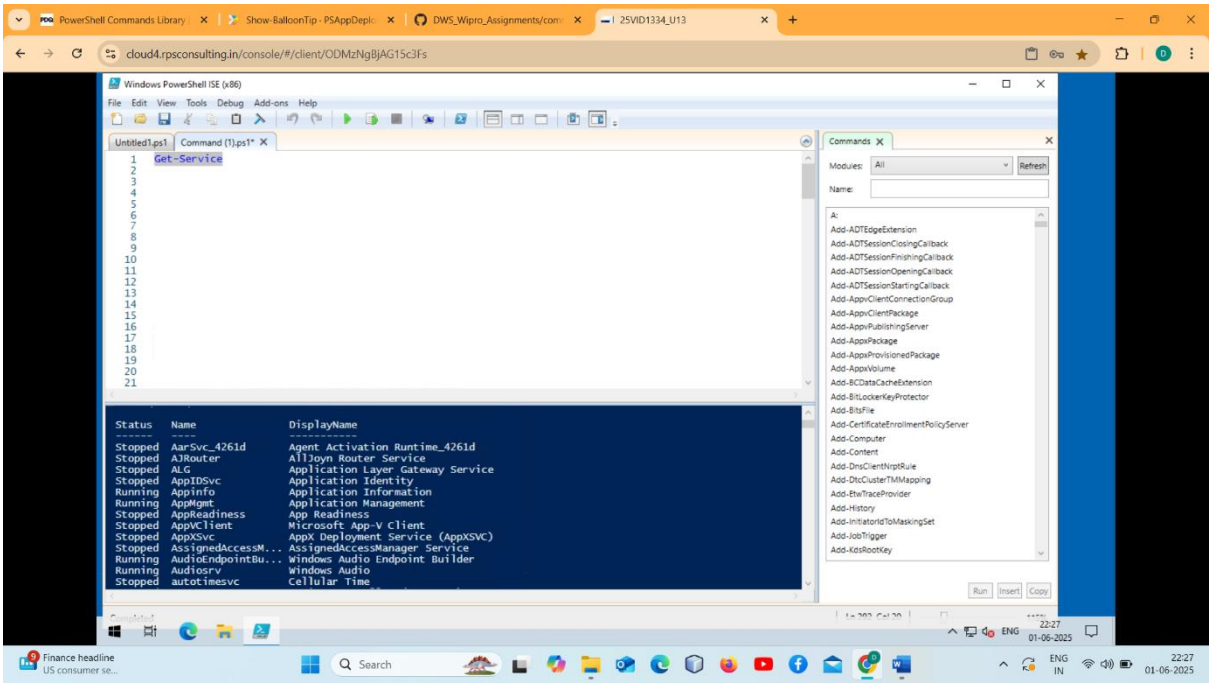
  Lists all available PowerShell commands, including cmdlets, functions,and aliases



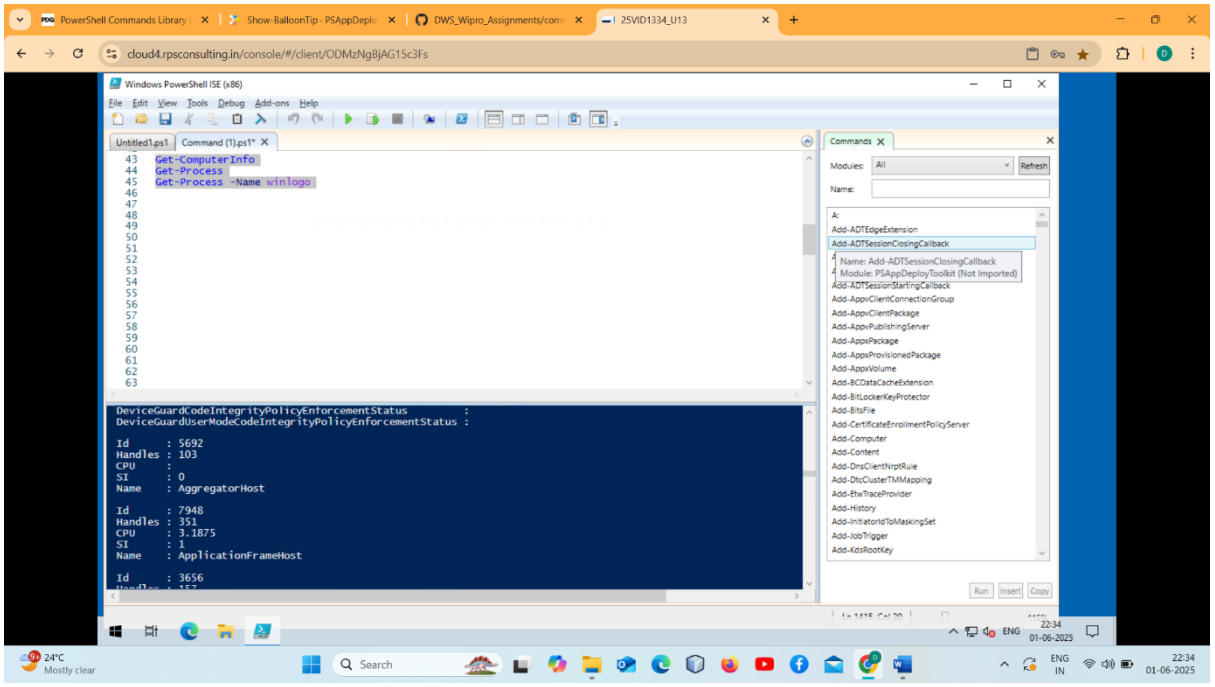
Show all commands

- **Get-Service**

  Gets the services on a local or remote computer



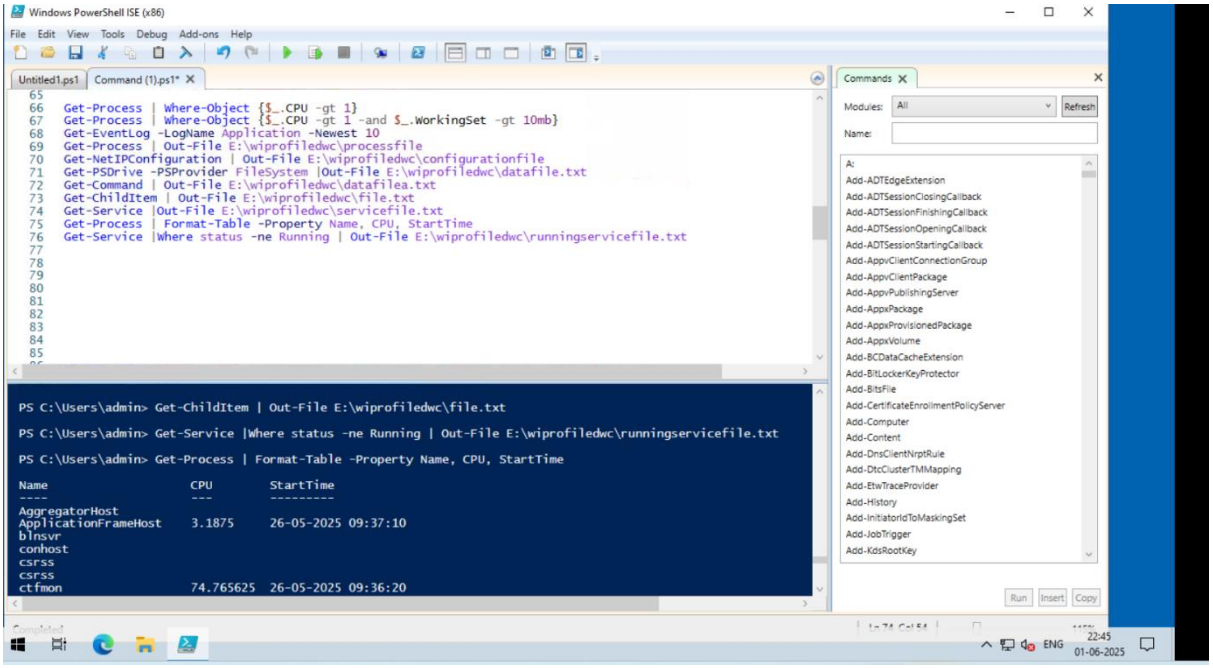
- **Get-ComputerInfo**
  Gets a consolidated object of system and operating system properties.
- **Get-Process**
  Retrieves information about running processes on the system
- **Get-Module**
  Gets the modules that have been imported or that can be imported into the current session.
- **Get-Process -Name winlogo**
  Retrives information of specific process like winlogo



- **Get-ExecutionPolicy**
  Gets the execution policies for the current session.
- **Get-Process | Where-Object {$_.CPU -gt 1}**
  This command shows only those running processes that have used more than 1 second of CPU time.
- **Get-Process | Where-Object {$_.CPU -gt 1 -and $_.WorkingSet -gt 10mb}**
  This command shows only the processes that are using more than 1 second of CPU time and more than 10 MB of memory.
- **Get-EventLog -LogName Application -Newest 10**
  This command displays the 10 latest events from the Application log
- **New-PSDrive -Name "Public" -PSProvider "FileSystem" -Root "\\Server01\Public"**
  This command creates a new PowerShell drive named "Public"

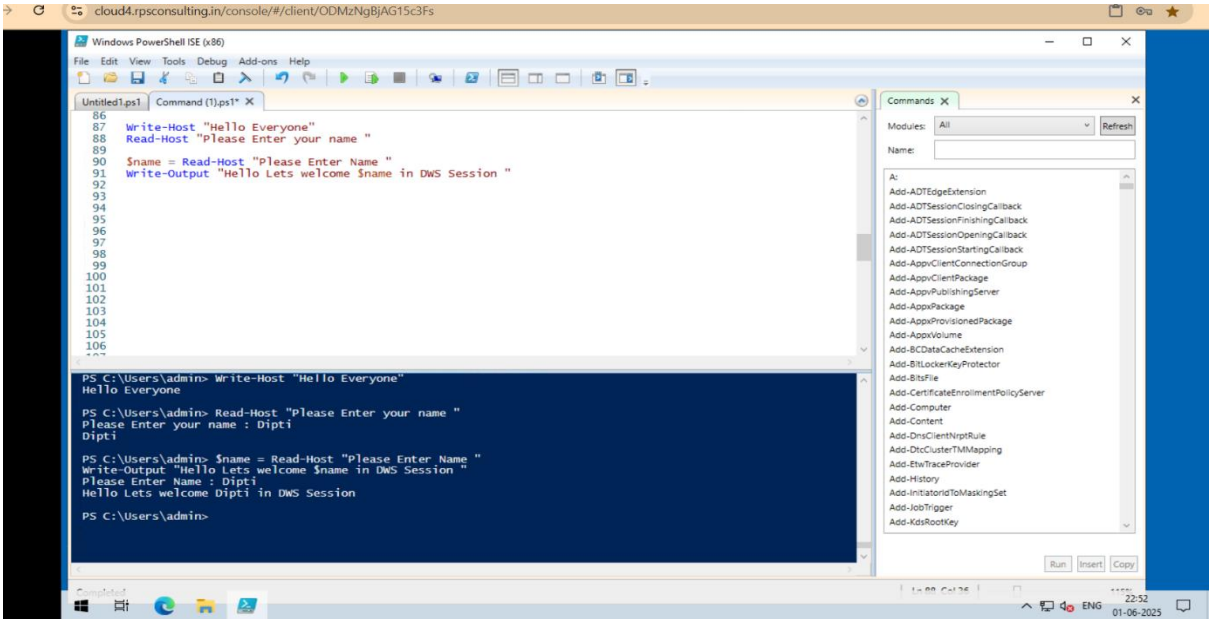
- **Get-Process | Out-File E:\wiprofiledwc\processfile**
  This command saves the list of running processes into a specified file
- **Get-NetIPConfiguration | Out-File E:\wiprofiledwc\configurationfile**
  This command collects your computer's network configuration details and saves them into given location file
- **Get-ChildItem | Out-File E:\wiprofiledwc\file.txt**

This command gets the list of files and folders in the current location and writes it to a text file.



- **Write-Host "Hello Everyone"**
  This command simply prints "Hello Everyone" on your PowerShell screen.
- **Read-Host "Please Enter your name "**
  This command asks the user to type their name and waits for the input
- **$name = Read-Host "Please Enter Name "**
  Write-Output "Hello Lets welcome $name in DWS Session "

The command asks for your name, then says: "Hello Lets welcome [YourName] in DWS Session"

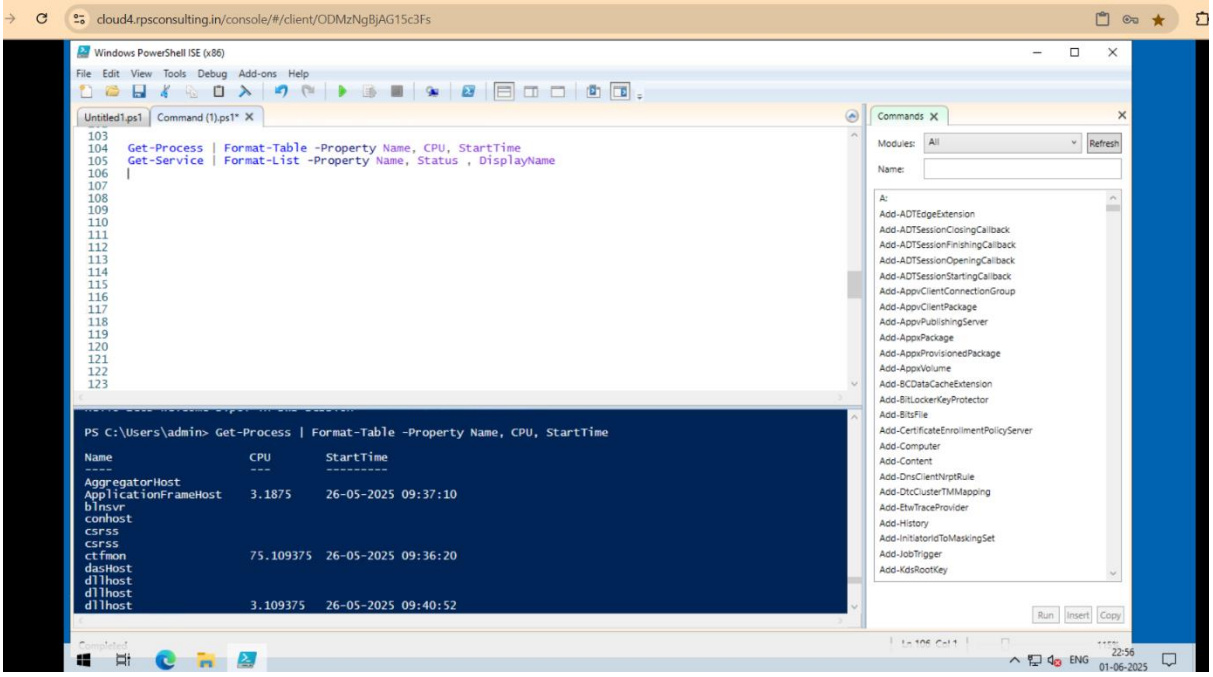


- **Format Table** : This command lists running processes and shows just their name, CPU time used, and when they started,in a table format

Get-Process | Format-Table -Property Name, CPU, StartTime

- **Format List** : This command lists all services, with service name, whether they are running or stopped, and display name, in a list.

Get-Service | Format-List -Property Name, Status , DisplayName



- **Pipeline** : This command finds all processes using more than 1 second of CPU, sorts them in descending order of CPU usage, and shows just their name and CPU time.

Get-Process | Where-Object {$_.CPU -gt 1 } | Sort-Object CPU -Descending | Select-Object Name , CPU

- **Conditional statement**

```
$age = 17
if($age -gt 18)
{
   Write-Host "Eligible to vote "
}
elseif($age -lt 18 )
{
   Write-Host "Not eligible to vote "
```

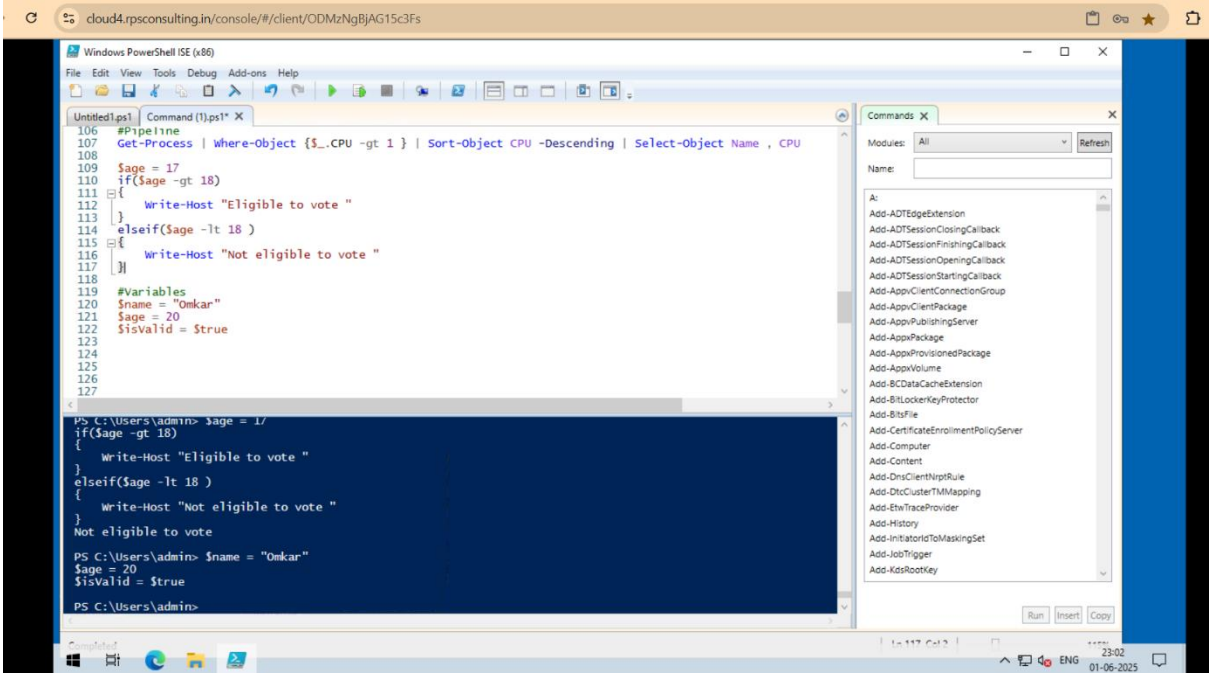
}

- **Variables**

  $name = "Omkar"

  $age = 20

  $isValid = $true



- **Array**

1. Creates an array named $myArray with 5 items:

   $myArray = 10,56,78,"Banana" ,"Mango "

2. Gets the first item in the array, which is 10.

   $myArray[0]

3. Gets the last item in the array, which is "Mango ".

   $myArray[-1]

4. Gets the second last item, which is "Banana".

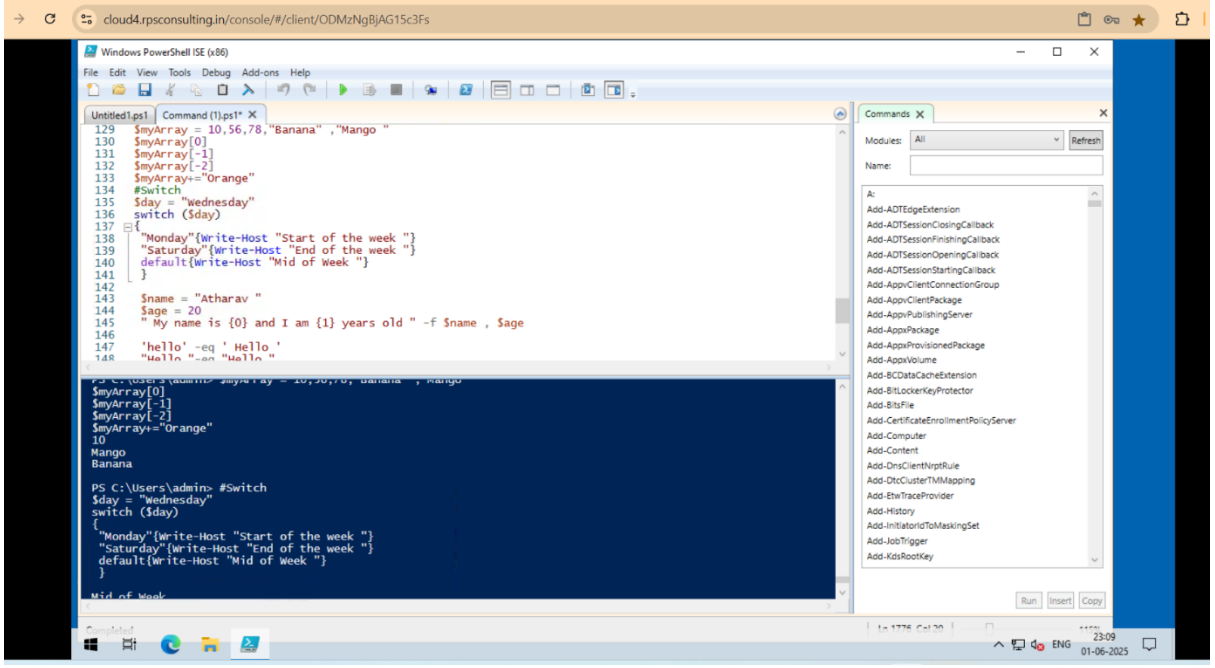
   $myArray[-2]

5. Adds the string "Orange" to the end of the array.

   $myArray+="Orange"

- **Switch** : The switch statement is used to test a value against multiple conditions and run matching code blocks

```
$day = "Wednesday"
switch ($day){
"Monday"{Write-Host "Start of the week "}
"Saturday"{Write-Host "End of the week "}
default{Write-Host "Mid of Week "}}
```

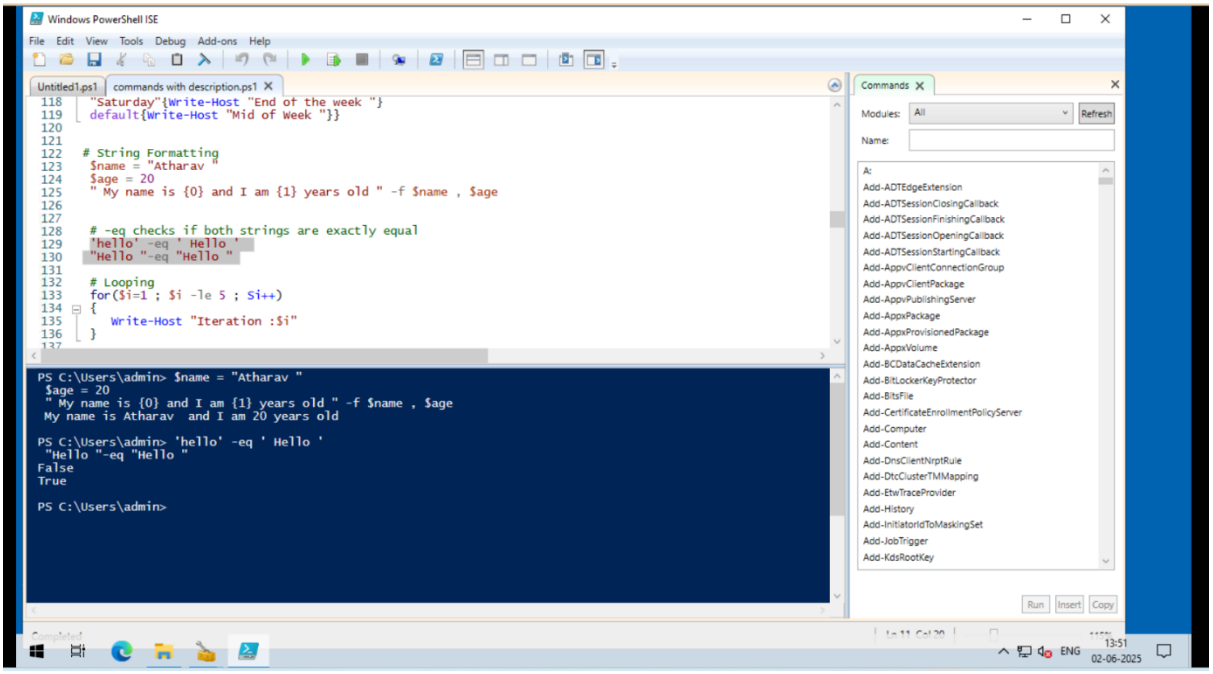


- **String Formatting**

```
$name = "Atharav "
$age = 20
" My name is {0} and I am {1} years old " -f $name , $age
```

- **-eq** checks if both strings are exactly equal

```
'hello' -eq ' Hello '
"Hello "-eq "Hello "
```



- **Execute MSI**

1. **Execute-MSI -Action 'Install' -Path 'Adobe_FlashPlayer_11.2.202.233_x64_EN.msi'**
   This command is used to install an MSI package
2. **Execute-MSI -Action 'Install' -Path 'Adobe_FlashPlayer_11.2.202.233_x64_EN.msi' -Transform 'Adobe_FlashPlayer_11.2.202.233_x64_EN_01.mst' -Parameters '/QN'**
   Installs an MSI, applying a transform and overriding the default MSI toolkit parameters
3. **Execute-MSI -Action 'Uninstall' -Path '{26923b43-4d38-484f-9b9e-de460746276c}'**
   Uninstalls an MSI using a product code
4. **Execute-MSI -Action 'Patch' -Path 'Adobe_Reader_11.0.3_EN.msp'**
   Installs an MSP
5. **Execute-MSI -Action Install -Path $AppMSIName -SkipMSIAlreadyInstalledCheck -ContinueOnError $False -LogName "${AppMSIName}_MSI"**
   This command installs an MSI application (whose path is stored in $AppMSIName), skips the "already installed" check, stops on errors, and saves a log file with the name format: AppName_MSI.log.

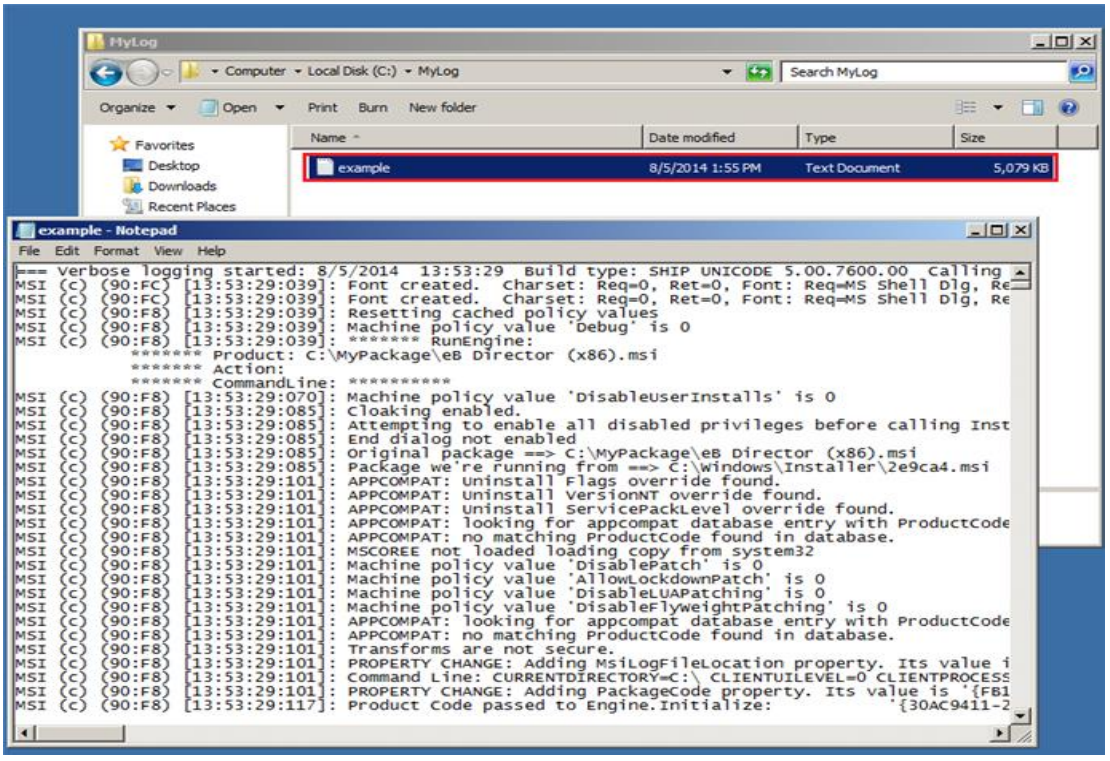
- **Execute Process:**

1. **Execute-Process -Path 'uninstall_flash_player_64bit.exe' -Parameters '/uninstall' -WindowStyle 'Hidden'** :
   If the file is in the "Files" directory of the App Deploy Toolkit, only the file name needs to be specified.
2. **Execute-Process -Path "$dirFiles\Bin\setup.exe" -Parameters '/S' -WindowStyle 'Hidden'**
3. **Execute-Process -Path 'setup.exe' -Parameters '/S' -IgnoreExitCodes '1,2'**
4. **Execute-Process -Path 'setup.exe' -Parameters "-s -f2`"$configToolkitLogDir\$installName.log`""**

5. **Execute-Process -Path 'setup.exe' -Parameters "/s /v`"ALLUSERS=1 /qn /L* `"$configToolkitLogDirr\$installName.log`"`""**

- **Execute MSP**

1. **Execute-MSP -Path 'Adobe_Reader_11.0.3_EN.msp'**
2. **Execute-MSP -Path 'AcroRdr2017Upd1701130143_MUI.msp' -AddParameters 'ALLUSERS=1'**



- **Copy File**

1. **Copy-File -Path "$dirSupportFiles*.*" -Destination "$envTemp\tempfiles"**
   Copy all of the files in a folder to a destination folder.
2. **Copy-File -Path "$dirSupportFiles\MyApp.ini" -Destination "$envWinDir\MyApp.ini"**
   Copy the specific mentioned files in a folder to a destination folder.

- **Execute MSUpdates**

1. **Install-MSUpdates -Directory "$dirFiles\MSUpdates"**
   This command installs all Microsoft update files (like .msu) from the folder

- **Set-ActiveSetup**

1. **Set-ActiveSetup -StubExePath 'C:\Users\Public\Company\ProgramUserConfig.vbs' -Arguments '/Silent' -Description 'Program User Config' -Key 'ProgramUserConfig' -Locale 'en'**

2. **Set-ActiveSetup -StubExePath "$envWinDir\regedit.exe" -Arguments "/S `"%SystemDrive%\Program Files (x86)\PS App Deploy\PSAppDeployHKCUSettings.reg`"" -Description 'PS App Deploy Config' -Key 'PS_App_Deploy_Config' -ContinueOnError $true**

- **Get-RegistryKey**

1. **Get-RegistryKey -Key 'HKLM:SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall\{1AD147D0BE0E-3D6C-AC11-64F6DC4163F1}'**
   Retrieves value names and value data for a specified registry key or optionally, a specific value.

- **Set INIValue**

1. **Set-IniValue -FilePath "C:\Windows\Path\to\my.ini" -Section 'MySection' -Key 'MyKey' -Value 'MyValue'**
   Opens an INI file and sets the value of the specified section and key.

- **Create/Delete/Modify Registries**

1. **New-Item -Path "HKLM:\Software\MyCompany" -Type Directory -ErrorAction SilentlyContinue**
   Create new key : This command creates a new registry key
2. **Remove-Item -Path "HKLM:\Software\MyCompany" -Recurse -Force -ErrorAction SilentlyContinue**
   Delete a Key : This command delete a registry key
3. **Set-ItemProperty -Path "HKLM:\Software\MyCompany\MyKey" -Name "MyValue" -Value "MyData"**
   Set a registry value :This command updates a registry value.
4. **Get-ItemProperty -Path "HKLM:\Software\MyCompany\MyKey" -Name "MyValue"**
   Get a registry value :This command gives a registry value.

- **Start/Stop Services**

1. Gets the services on a local or remote computer.

   Start Service

2. Start the service having name YourServiceName

   Start-Service -Name "YourServiceName"

   Start-Service -Name "wuauserv".

3. This Command Stop the service

   Stop Service

4. This Command stop the specified name service

   Stop-Service -Name "YourServiceName"

   Stop-Service -Name "wuauserv"

5. This Command stop the service if it has a dependent service

   Stop-Service -Name "YourServiceName" -Force

- **Start Sevice with dependencies**

1. Start service use the Start-ServiceAndDependencies function to start a service and dependent service

   Start-ServiceAndDependencies -Name "YourServiceName"

   Start-ServiceAndDependencies -Name "wuauserv"

- **Stop Service with dependencies**

1. Stop service use the Stop-ServiceAndDependencies function to stop a service and dependent service

   Stop-ADTServiceAndDependencies -Name "YourServiceName"

   Stop-ADTServiceAndDependencies -Name "wuauserv"

- **Show-InstallationPrompt**

1. Displays a custom installation prompt with the toolkit branding and optional buttons.

   Show-InstallationPrompt -Message 'Do you want to proceed with the installation?' -ButtonRightText 'Yes' -ButtonLeftText 'No'

Log Text	Component	Date/Time	Thread
[Installation] :: Found no application based on the supplied parameters.	Get-InstalledApplication	24/04/2022 10:41:55	9632 (0x25A0)
[Installation] :: Executing MSI action [Install]...	Execute-MSI	24/04/2022 10:41:55	9632 (0x25A0)
[Installation] :: [C:\WINDOWS\System32\msiexec.exe] is a valid fully qualified path, continue.	Execute-Process	24/04/2022 10:41:55	9632 (0x25A0)
[Installation] :: Check to see if mutex [Global_MSIExecute] is available. Wait up to [10 minute(s)] for the mutex to becom...	Test-IsMutexAvailable	24/04/2022 10:41:55	9632 (0x25A0)
[Installation] :: Mutex [Global_MSIExecute] is available for an exclusive lock.	Test-IsMutexAvailable	24/04/2022 10:41:55	9632 (0x25A0)
[Installation] :: Working Directory is [C:\WINDOWS\ccmcache\k\Files].	Execute-Process	24/04/2022 10:41:56	9632 (0x25A0)
[Installation] :: Executing [C:\WINDOWS\System32\msiexec.exe /i "C:\WINDOWS\ccmcache\k\Files\ImprivataAgent_x64....	Execute-Process	24/04/2022 10:41:56	9632 (0x25A0)
[Installation] :: PassThru parameter specified, returning execution results object.	Execute-Process	24/04/2022 10:43:22	9632 (0x25A0)
[Installation] :: Execution completed successfully with exit code [3010]. A reboot is required.	Execute-Process	24/04/2022 10:43:22	9632 (0x25A0)
[Installation] :: Refresh the Desktop and the Windows Explorer environment process block.	Update-Desktop	24/04/2022 10:43:22	9632 (0x25A0)
[Post-Installation] :: Set registry key value: [Registry::HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\Curren...	Set-RegistryKey	24/04/2022 10:43:22	9632 (0x25A0)
[Post-Installation] :: Show-InstallationRestartPrompt was invoked, but an existing restart prompt was detected. Cancellin...	Show-InstallationRestartPrc	24/04/2022 10:43:22	9632 (0x25A0)
[Post-Installation] :: Close the installation progress dialog.	Close-InstallationProgress	24/04/2022 10:43:22	9632 (0x25A0)
[Post-Installation] :: [redacted] Installation completed with exit code [0].	Exit-Script	24/04/2022 10:43:22	9632 (0x25A0)
[Post-Installation] :: Check if PowerPoint is in either fullscreen slideshow mode or presentation mode...	Test-PowerPoint	24/04/2022 10:43:22	9632 (0x25A0)
[Post-Installation] :: PowerPoint application is not running.	Test-PowerPoint	24/04/2022 10:43:22	9632 (0x25A0)
[Post-Installation] :: PowerPoint is running in fullscreen mode [False].	Test-PowerPoint	24/04/2022 10:43:22	9632 (0x25A0)
[Post-Installation] :: Display balloon tip notification asynchronously with message [Installation complete.].	Show-BalloonTip	24/04/2022 10:43:22	9632 (0x25A0)
[Post-Installation] :: [C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe] is a valid fully qualified path, conti...	Execute-Process	24/04/2022 10:43:22	9632 (0x25A0)
[Post-Installation] :: Working Directory is [C:\Windows\System32\WindowsPowerShell\v1.0].	Execute-Process	24/04/2022 10:43:22	9632 (0x25A0)
[Post-Installation] :: Executing [C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe [PowerShell ScriptBlock]...	Execute-Process	24/04/2022 10:43:22	9632 (0x25A0)
[Post-Installation] :: NoWait parameter specified. Continuing without waiting for exit code...	Execute-Process	24/04/2022 10:43:22	9632 (0x25A0)
[Post-Installation] :: ----------	Exit-Script	24/04/2022 10:43:22	9632 (0x25A0)

- **Show-InstallationWelcome**

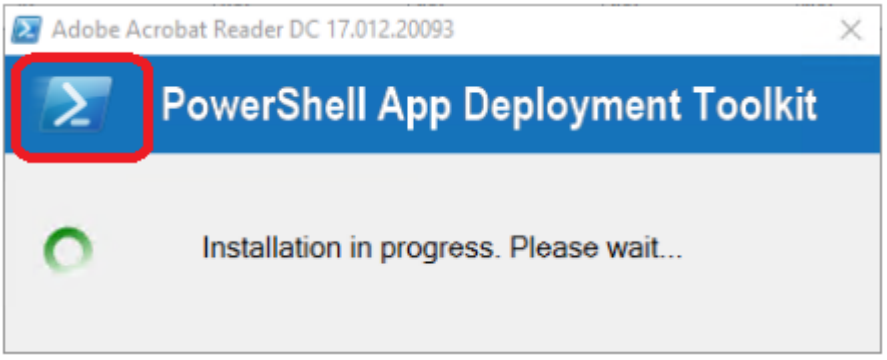
1. Prompt the user to close Internet Explorer, Word and Excel.

Show-InstallationWelcome -CloseApps 'iexplore,winword,excel'

- **Show-InstallationProgress**

1. Uses the default status message from the XML configuration file.

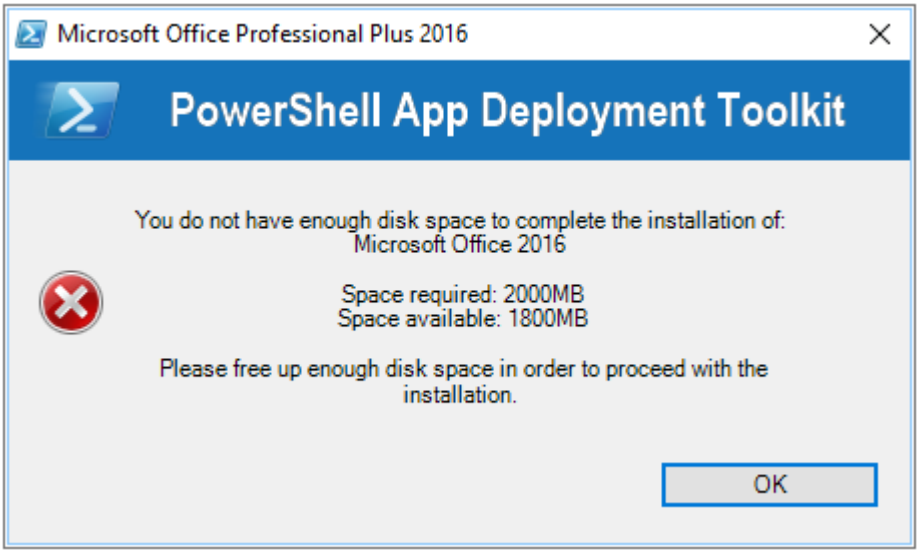
Show-InstallationProgress



- **Show-DialogBox**

1. Display a custom dialog box with optional title, buttons, icon and timeout.

Show-DialogBox -Title 'Installed Complete' -Text 'Installation has completed. Please click OK and restart your computer.' -Icon 'Information'



- **Show-BallonTip**

1. Displays a balloon tip notification in the system tray.

Show-BalloonTip -BalloonTipText Updates' -BalloonTipTitle 'WindowsPowerShell ISE'