Project Title:

PSADT DEPLOYMENTS ON vM & running/modifying script based on requirement.

Created By:-

Krishna Verma

User id- 34851

Email - kv979539g@gmail.com

Project Overview :

The goal of our project is to learn how to create, test, and modify a software deployment package using the PowerShell App Deployment Toolkit (PSADT). We will focus on creating a wrapper for a standard application installer (such as an .msi file), testing its functionality on a virtual machine (VM), and understanding how to modify the script to meet specific requirements.

Key Concepts :

- **PowerShell App Deployment Toolkit (PSADT):** It is an open-source framework that standardizes the process of deploying Windows applications. It provides a robust, pre-built PowerShell script and a set of functions to handle common deployment tasks.
- **Wrapper:** A script that "wraps" around an application's native installer, providing additional functionality like user prompts, custom actions (e.g., creating a file or registry key), and detailed logging.
- **Virtual Machine (VM):** An essential tool for testing. A VM allows you to simulate a clean, real-world user environment without affecting your primary operating system.

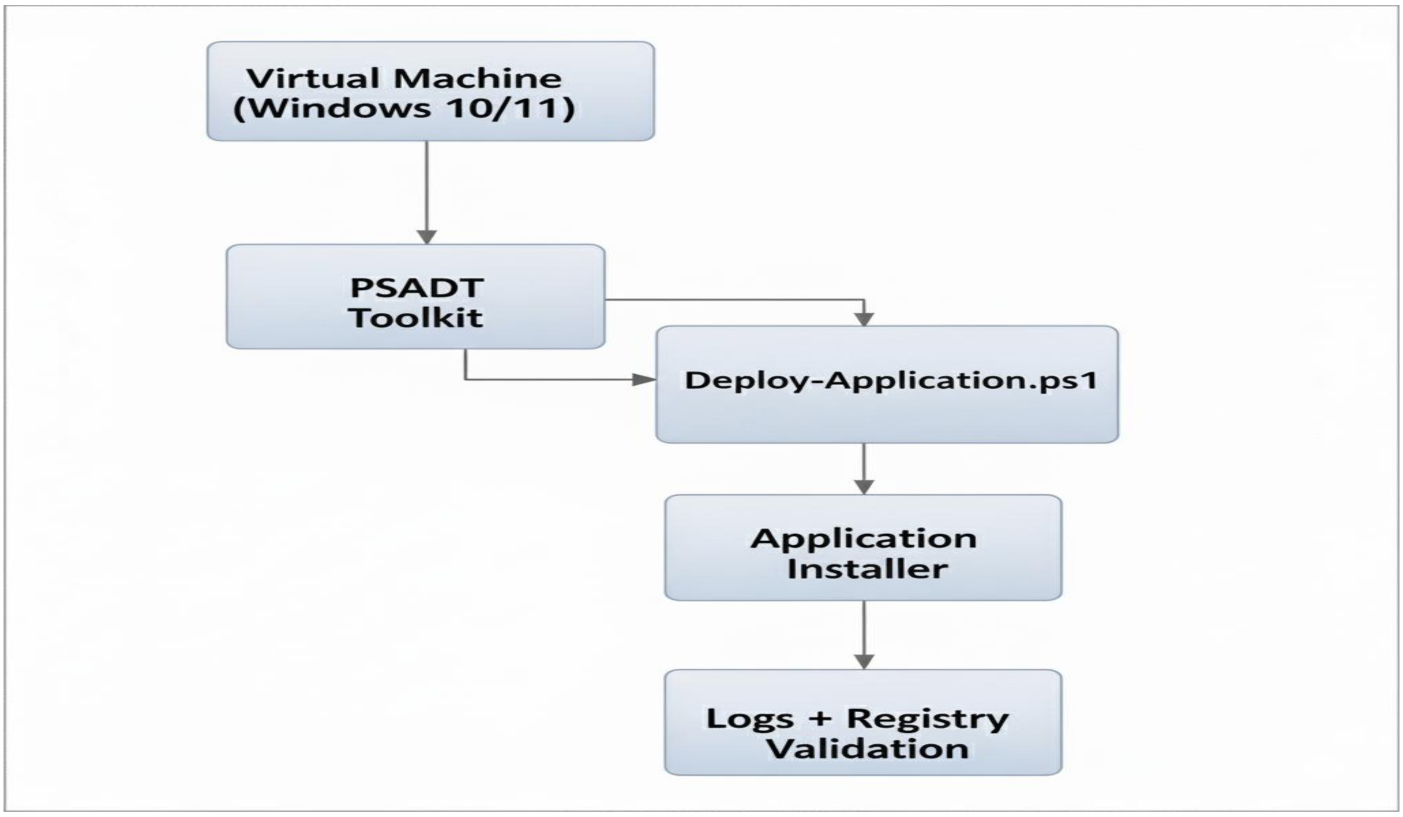
Objectives :

- Automate the deployment of applications using PSADT.
- Provide a standardized install/uninstall mechanism.
- Demonstrate deployment in a controlled VM environment.
- Generate logs and registry keys for auditing.
- Gain hands-on experience in scripting and using PSADT.

Project Requirements :

- Windows 10/11 Virtual Machine
- PowerShell 5.1 or later
- PowerShell App Deployment Toolkit (PSADT)
- Application installer (MSI/EXE of our choice)
- Text Editor (VS Code / PowerShell ISE/ Notepad)
- Admin privileges on the VM

Architecture Diagram :



Execution Overview :

Step 1: Initial Setup

➢ **Download PSADT**

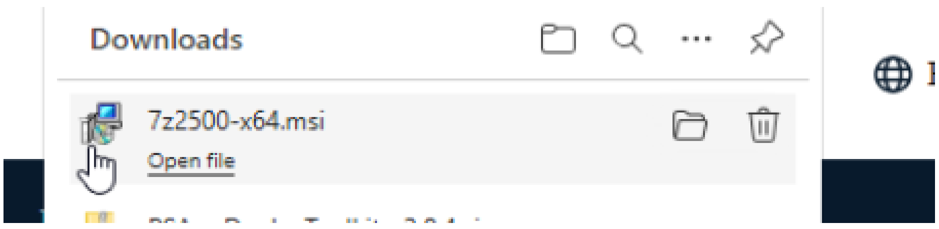
The first step is to download the latest version of the PSADT from its official GitHub repository.

- Go to the [PSADT GitHub Releases page]() and download the .zip file for the latest release.
- Extract the contents of the .zip file to a new, dedicated folder. This extracted folder will be our project's root directory. The most important files are Deploy-Application.ps1 and AppDeployToolkitMain.ps1.

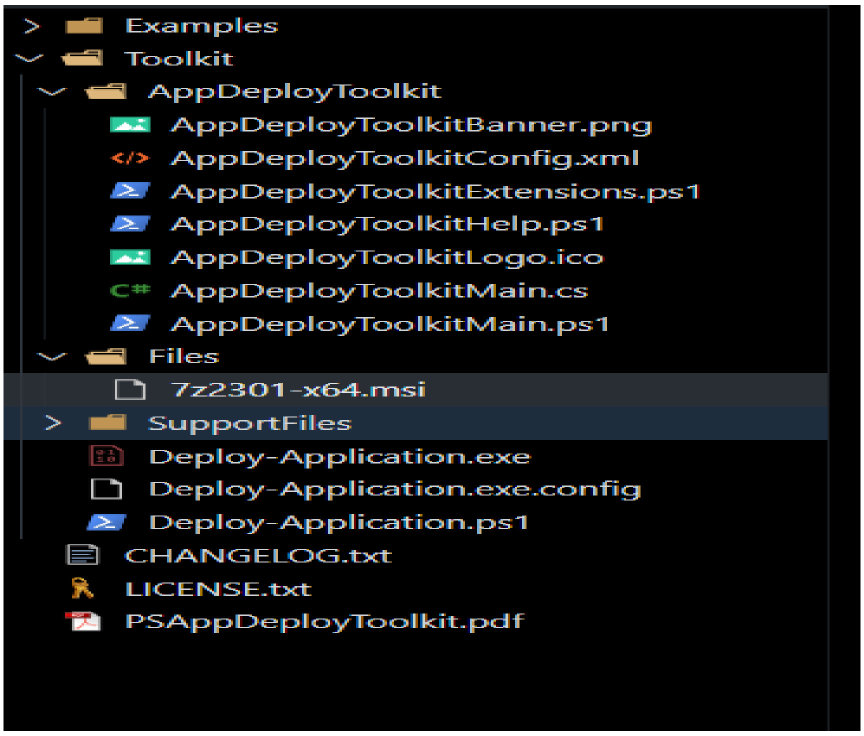
Step 2: Set up the Project Folder

Within the extracted PSADT folder, we will find a Files folder. This is where we will place the application installer that we want to deploy.

- **Example:** For our project, we use a sample application installer. Download the 64-bit MSI for [7-Zip]() and place it inside the Files folder.



Our project structure should look something like this:



```
> Examples
v Toolkit
  v AppDeployToolkit
      AppDeployToolkitBanner.png
      AppDeployToolkitConfig.xml
      AppDeployToolkitExtensions.ps1
      AppDeployToolkitHelp.ps1
      AppDeployToolkitLogo.ico
      AppDeployToolkitMain.cs
      AppDeployToolkitMain.ps1
  v Files
      7z2301-x64.msi
  > SupportFiles
    Deploy-Application.exe
    Deploy-Application.exe.config
    Deploy-Application.ps1
  CHANGELOG.txt
  LICENSE.txt
  PSAppDeployToolkit.pdf
```

Step 3: Prepare the Virtual Machine

- Set up a clean VM running Windows 10 or 11.

- Ensure PowerShell is updated to at least version 5.0.

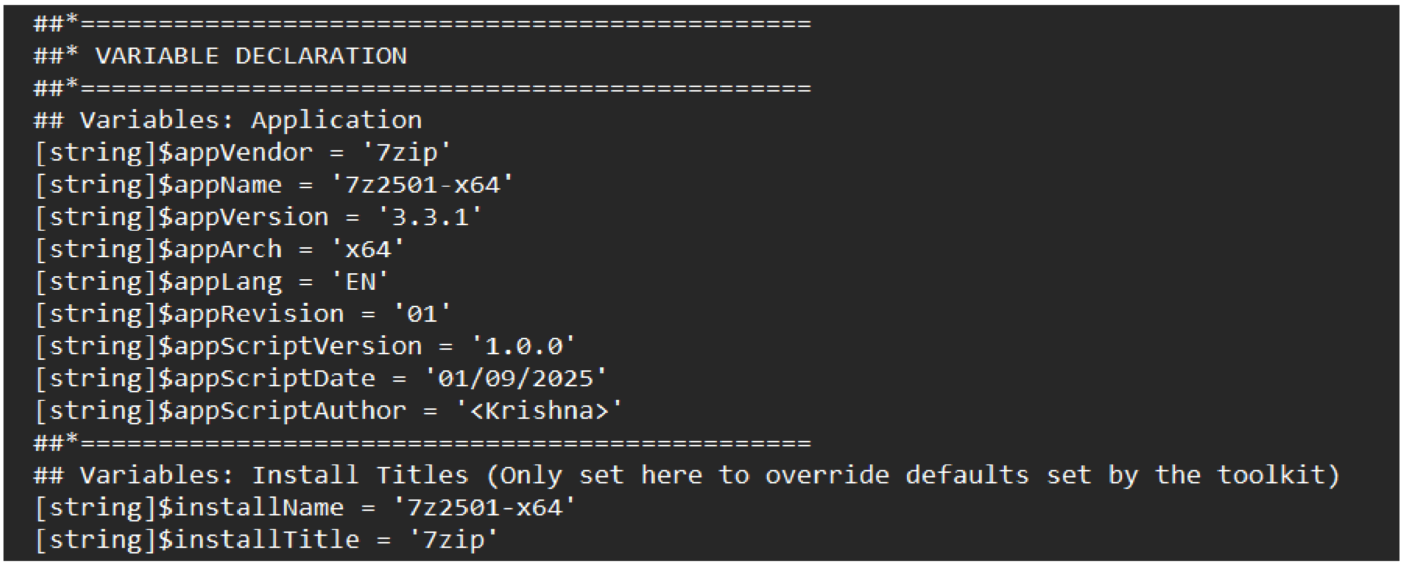
Step 4: Modifying the PSADT Script (Deploy-Application.ps1)

The Deploy-Application.ps1 script is the main file we will modify. It's pre-populated with comments and examples to take as a reference guide. We will work within specific sections of this script.

a. Fill Out Application Details :

At the top of the script, we will find a "Variable Declaration" section. We have to fill the details of the application that we need to deploy.

VARIABLE DECLARATION

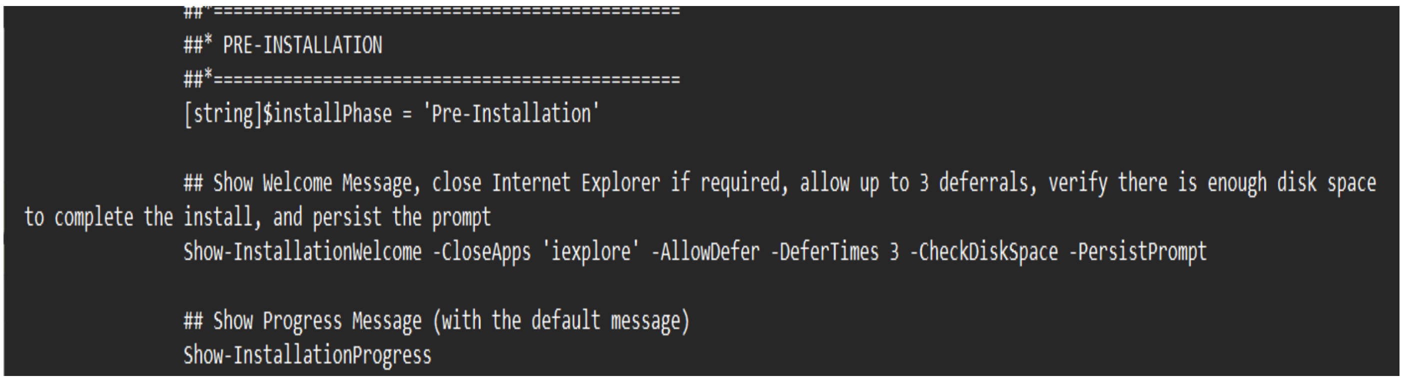


```
##*===============================================
##* VARIABLE DECLARATION
##*===============================================
## Variables: Application
[string]$appVendor = '7zip'
[string]$appName = '7z2501-x64'
[string]$appVersion = '3.3.1'
[string]$appArch = 'x64'
[string]$appLang = 'EN'
[string]$appRevision = '01'
[string]$appScriptVersion = '1.0.0'
[string]$appScriptDate = '01/09/2025'
[string]$appScriptAuthor = '<Krishna>'
##*===============================================
## Variables: Install Titles (Only set here to override defaults set by the toolkit)
[string]$installName = '7z2501-x64'
[string]$installTitle = '7zip'
```

b. Add Installation Logic :

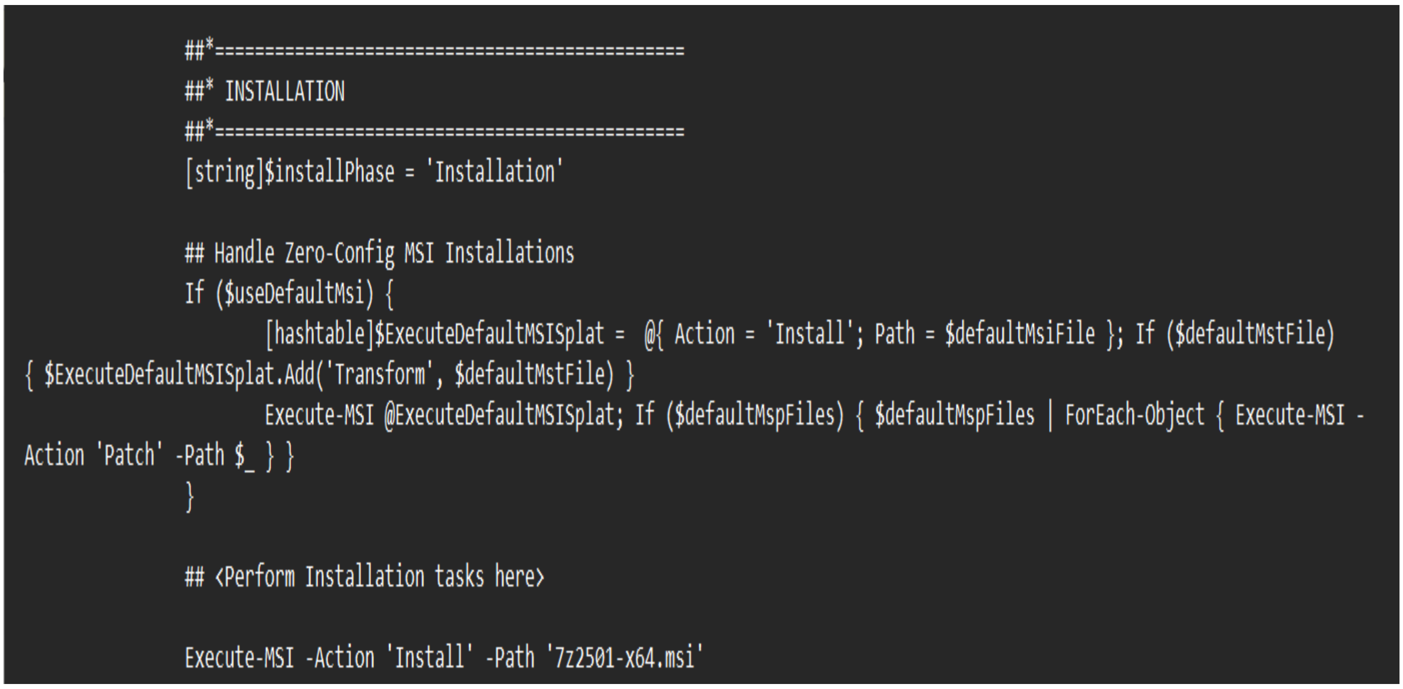
Scroll down to the Installation section. This is where we will add the commands to install our application. The PSADT framework provides a function specifically for MSI installers: Execute-MSI.

## Pre-Installation



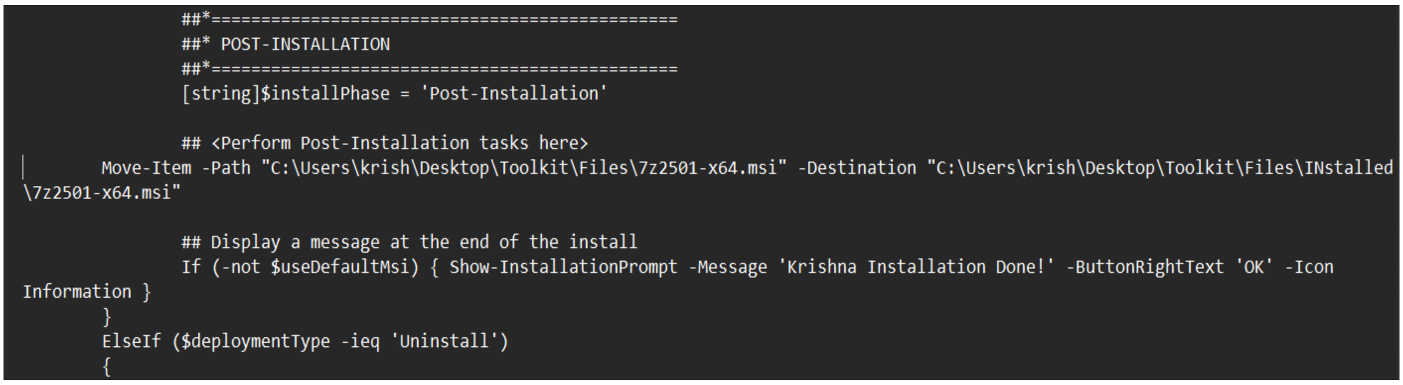
```
		##*===============================================
		##* PRE-INSTALLATION
		##*===============================================
		[string]$installPhase = 'Pre-Installation'

		## Show Welcome Message, close Internet Explorer if required, allow up to 3 deferrals, verify there is enough disk space
to complete the install, and persist the prompt
		Show-InstallationWelcome -CloseApps 'iexplore' -AllowDefer -DeferTimes 3 -CheckDiskSpace -PersistPrompt

		## Show Progress Message (with the default message)
		Show-InstallationProgress
```

## Installation



```
		##*===============================================
		##* INSTALLATION
		##*===============================================
		[string]$installPhase = 'Installation'

		## Handle Zero-Config MSI Installations
		If ($useDefaultMsi) {
			[hashtable]$ExecuteDefaultMSISplat =  @{ Action = 'Install'; Path = $defaultMsiFile }; If ($defaultMstFile)
{ $ExecuteDefaultMSISplat.Add('Transform', $defaultMstFile) }
			Execute-MSI @ExecuteDefaultMSISplat; If ($defaultMspFiles) { $defaultMspFiles | ForEach-Object { Execute-MSI -
Action 'Patch' -Path $_ } }
		}

		## <Perform Installation tasks here>

		Execute-MSI -Action 'Install' -Path '7z2501-x64.msi'
```

## Post-Installation

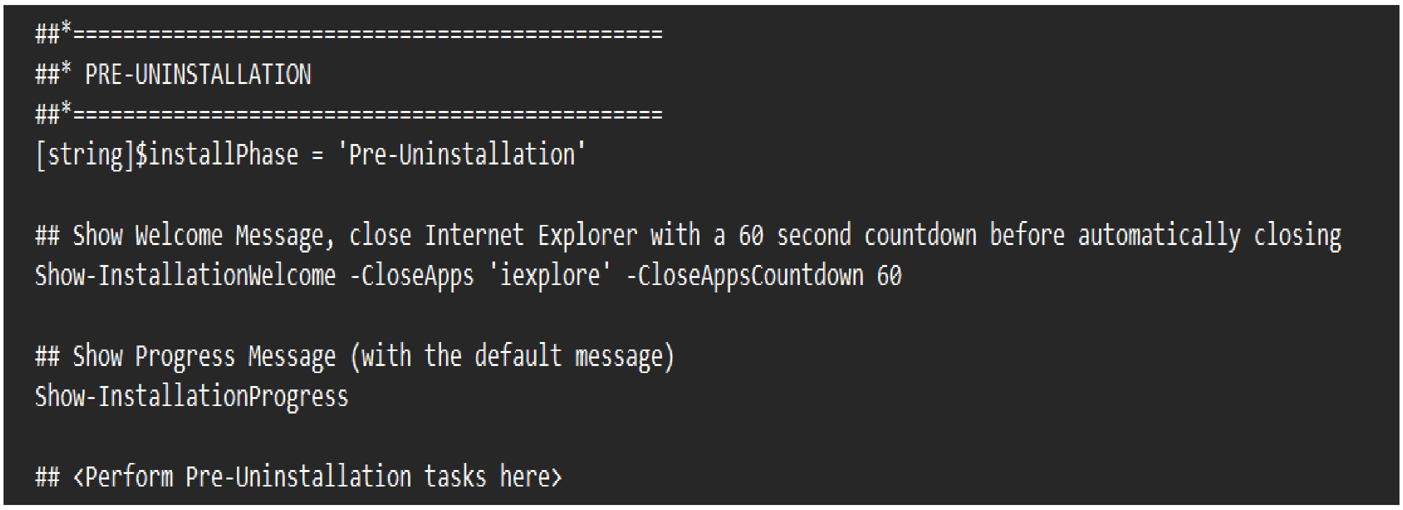


```
		##*===============================================
		##* POST-INSTALLATION
		##*===============================================
		[string]$installPhase = 'Post-Installation'

		## <Perform Post-Installation tasks here>
	Move-Item -Path "C:\Users\krish\Desktop\Toolkit\Files\7z2501-x64.msi" -Destination "C:\Users\krish\Desktop\Toolkit\Files\INstalled
\7z2501-x64.msi"

		## Display a message at the end of the install
		If (-not $useDefaultMsi) { Show-InstallationPrompt -Message 'Krishna Installation Done!' -ButtonRightText 'OK' -Icon
Information }
	}
	ElseIf ($deploymentType -ieq 'Uninstall')
	{
```

c. Add Uninstallation Logic :

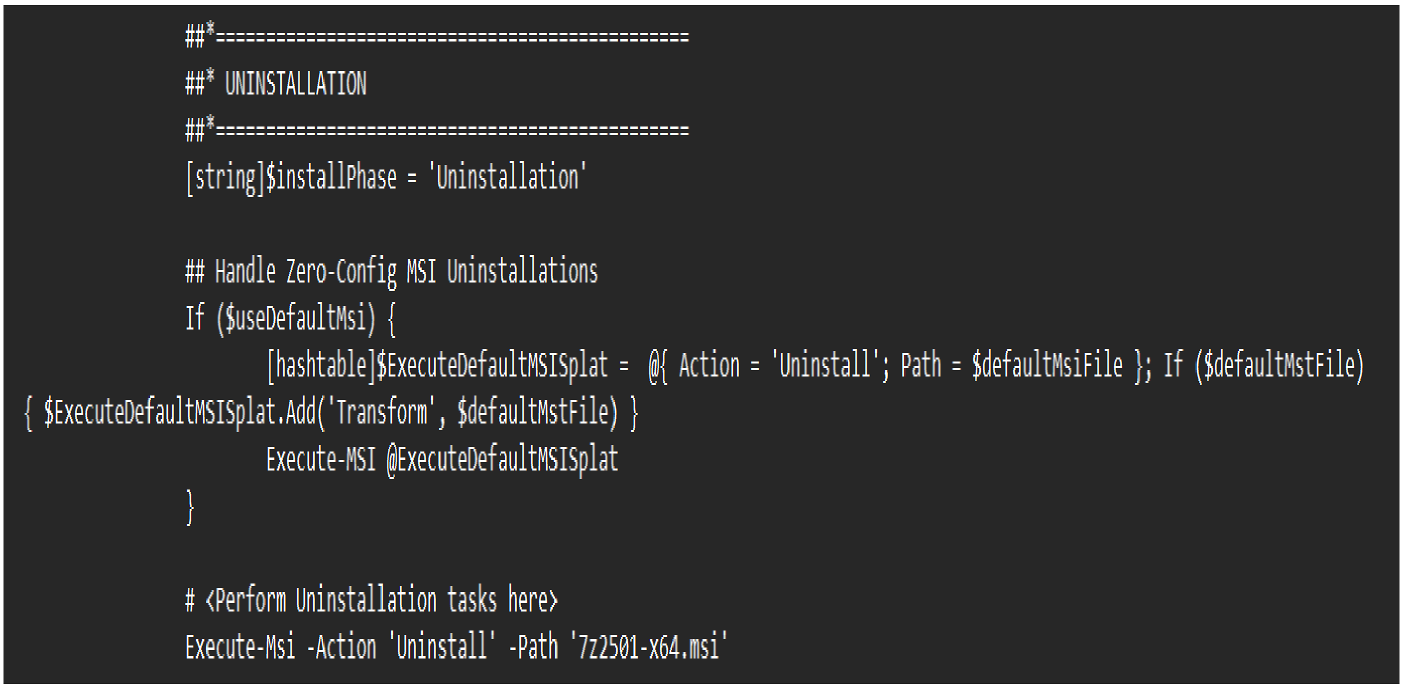
Scroll to the Uninstallation section. It's just as important to define how the application is removed. You will use the Execute-MSI function with the Uninstall action. The script can automatically find the MSI by its product code, or you can specify the MSI name as we did for the installation.

## Pre-Uninstallation



```
##*===============================================
##* PRE-UNINSTALLATION
##*===============================================
[string]$installPhase = 'Pre-Uninstallation'

## Show Welcome Message, close Internet Explorer with a 60 second countdown before automatically closing
Show-InstallationWelcome -CloseApps 'iexplore' -CloseAppsCountdown 60

## Show Progress Message (with the default message)
Show-InstallationProgress

## <Perform Pre-Uninstallation tasks here>
```

## Uninstallation



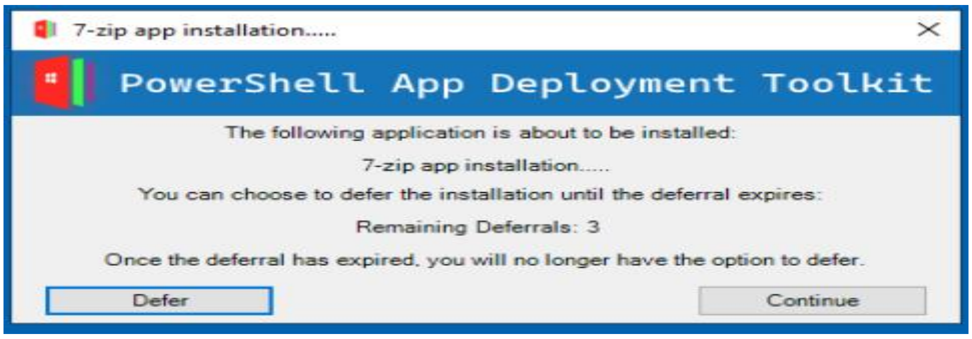
```
		##*===============================================
		##* UNINSTALLATION
		##*===============================================
		[string]$installPhase = 'Uninstallation'

		## Handle Zero-Config MSI Uninstallations
		If ($useDefaultMsi) {
			[hashtable]$ExecuteDefaultMSISplat =  @{ Action = 'Uninstall'; Path = $defaultMsiFile }; If ($defaultMstFile)
{ $ExecuteDefaultMSISplat.Add('Transform', $defaultMstFile) }
			Execute-MSI @ExecuteDefaultMSISplat
		}

		# <Perform Uninstallation tasks here>
		Execute-Msi -Action 'Uninstall' -Path '7z2501-x64.msi'
```

## Post-Uninstallation

```
            ##*===============================================
            ##* POST-UNINSTALLATION
            ##*===============================================
            [string]$installPhase = 'Post-Uninstallation'

            ## <Perform Post-Uninstallation tasks here>
```

Step 5 – Run Deployment :-

Installation :


7-zip app installation.....

PowerShell App Deployment Toolkit

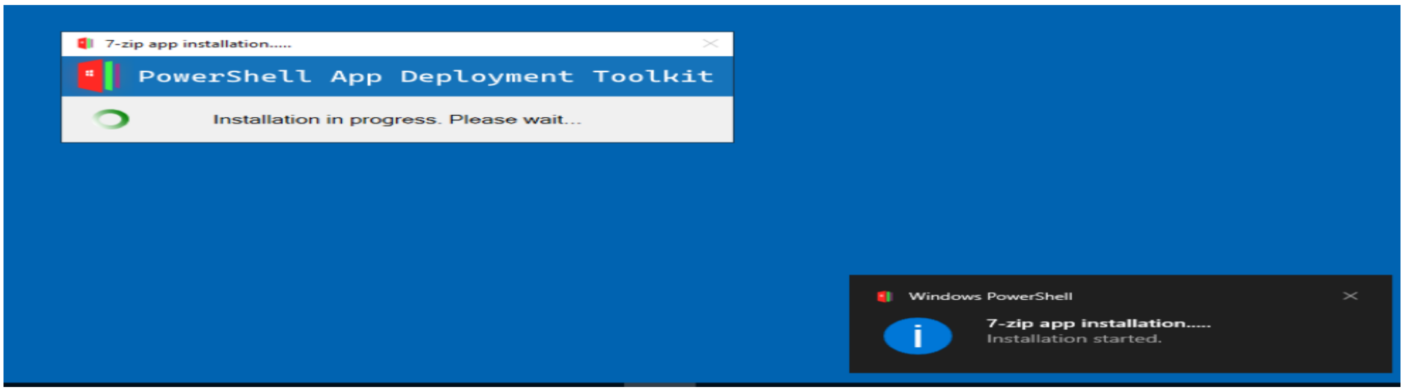
The following application is about to be installed:

7-zip app installation.....

You can choose to defer the installation until the deferral expires:

Remaining Deferrals: 3

Once the deferral has expired, you will no longer have the option to defer.


7-zip app installation.....

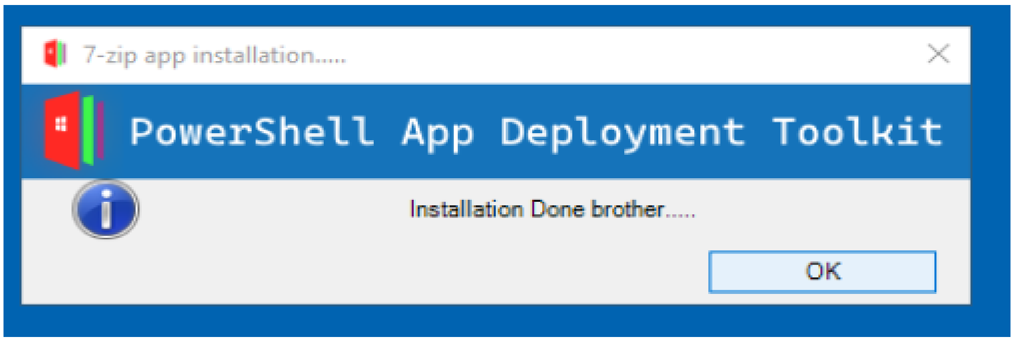
PowerShell App Deployment Toolkit

Installation in progress. Please wait...

Windows PowerShell

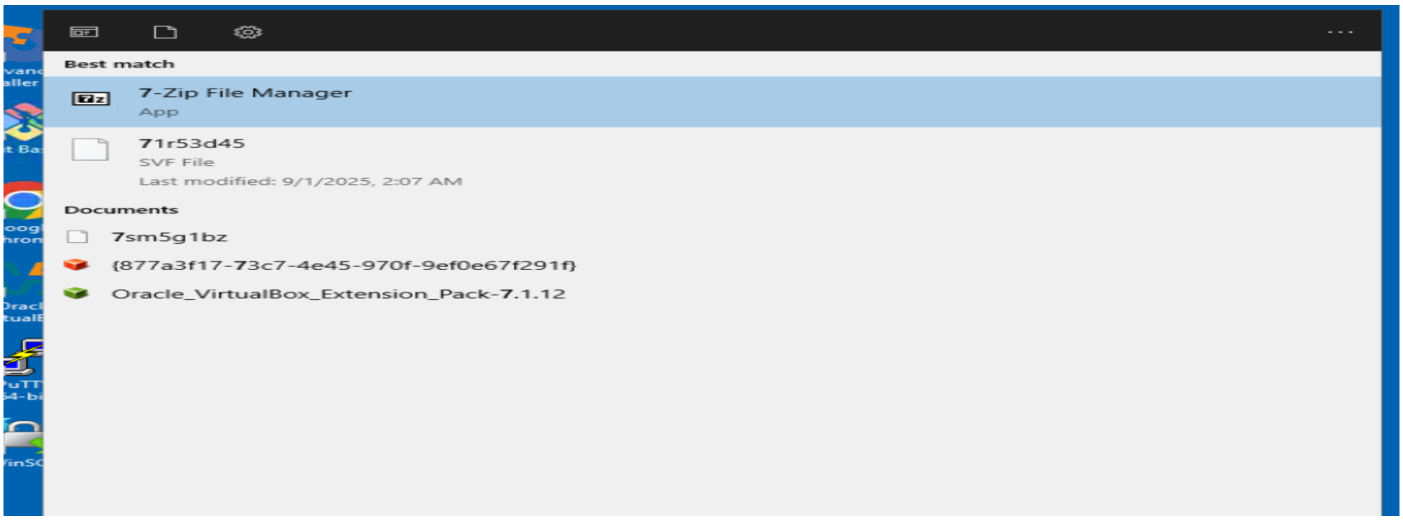
7-zip app installation.....

Installation started.


7-zip app installation.....

PowerShell App Deployment Toolkit

Installation Done brother.....


Best match

7-Zip File Manager
App

71r53d45
SVF File
Last modified: 9/1/2025, 2:07 AM

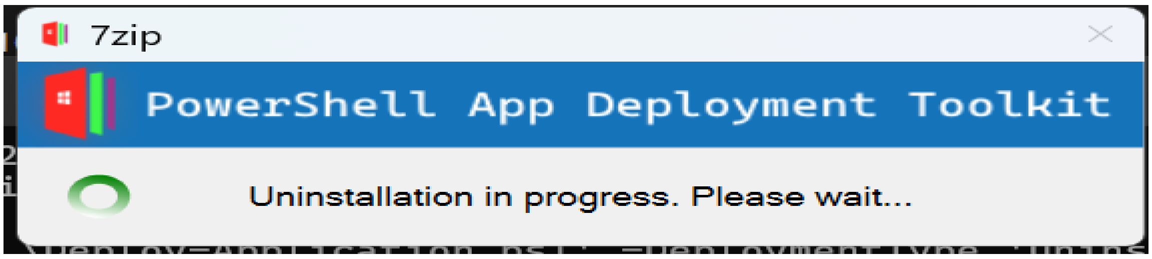
Documents

7sm5g1bz

{877a3f17-73c7-4e45-970f-9ef0e67f291f}

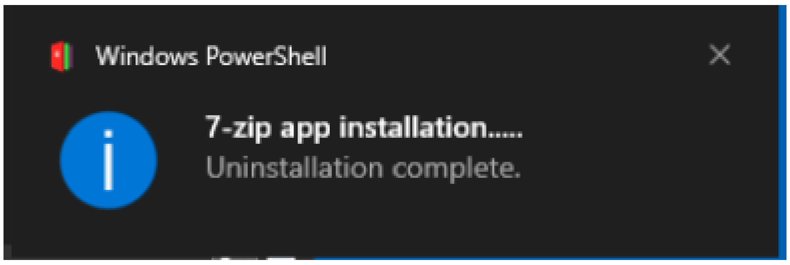
Oracle_VirtualBox_Extension_Pack-7.1.12

Uninstallation :-


7zip

PowerShell App Deployment Toolkit

Uninstallation in progress. Please wait...


Windows PowerShell

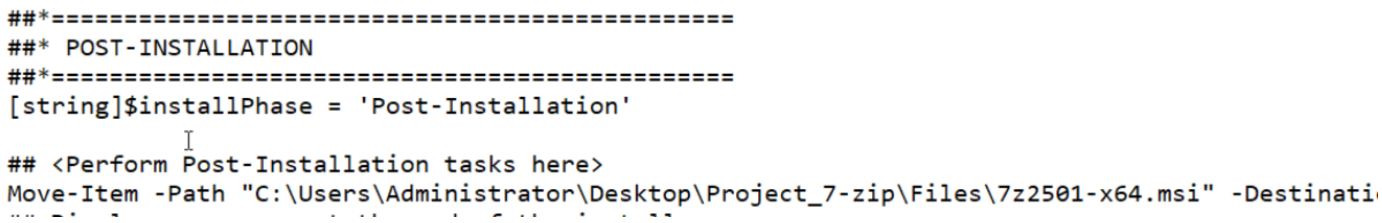
7-zip app installation.....

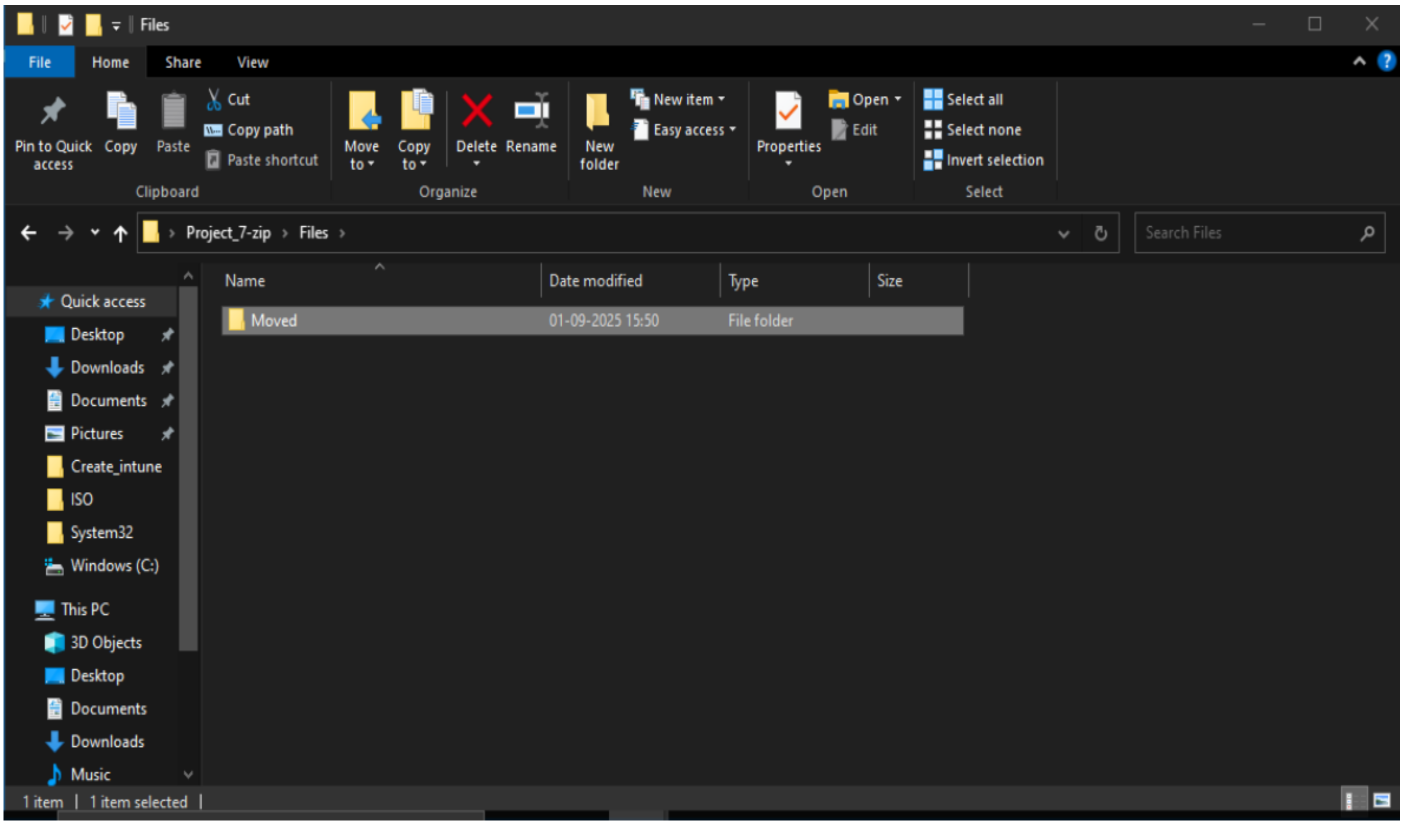
Uninstallation complete.

Modifying Script :-

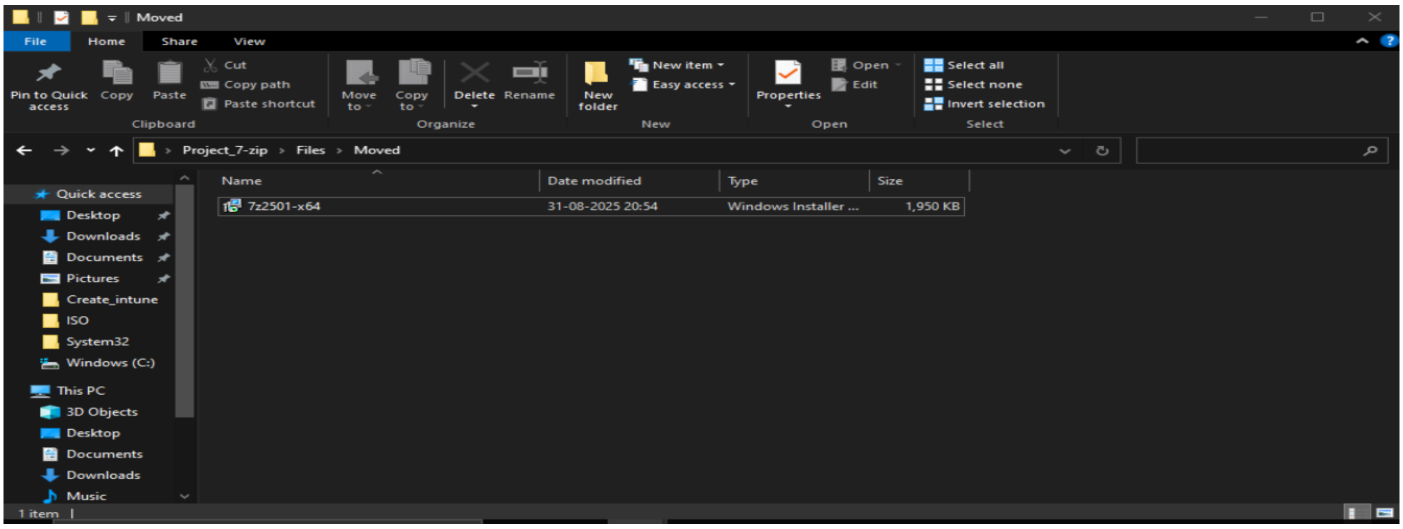
- We need to move folder after installation into another folder.
- Then we need to modify the Post installation script.
- In this task me moved our installer .msi file into a new folder during the post installation phase.
- The Script was like that in post-installation section:

  Move-Item -Path “” -Destination “” -Force


```
##*===============================================
##* POST-INSTALLATION
##*===============================================
[string]$installPhase = 'Post-Installation'

## <Perform Post-Installation tasks here>
Move-Item -Path "C:\Users\Administrator\Desktop\Project_7-zip\Files\7z2501-x64.msi" -Destinati
```



Name	Date modified	Type	Size
Moved	01-09-2025 15:50	File folder	



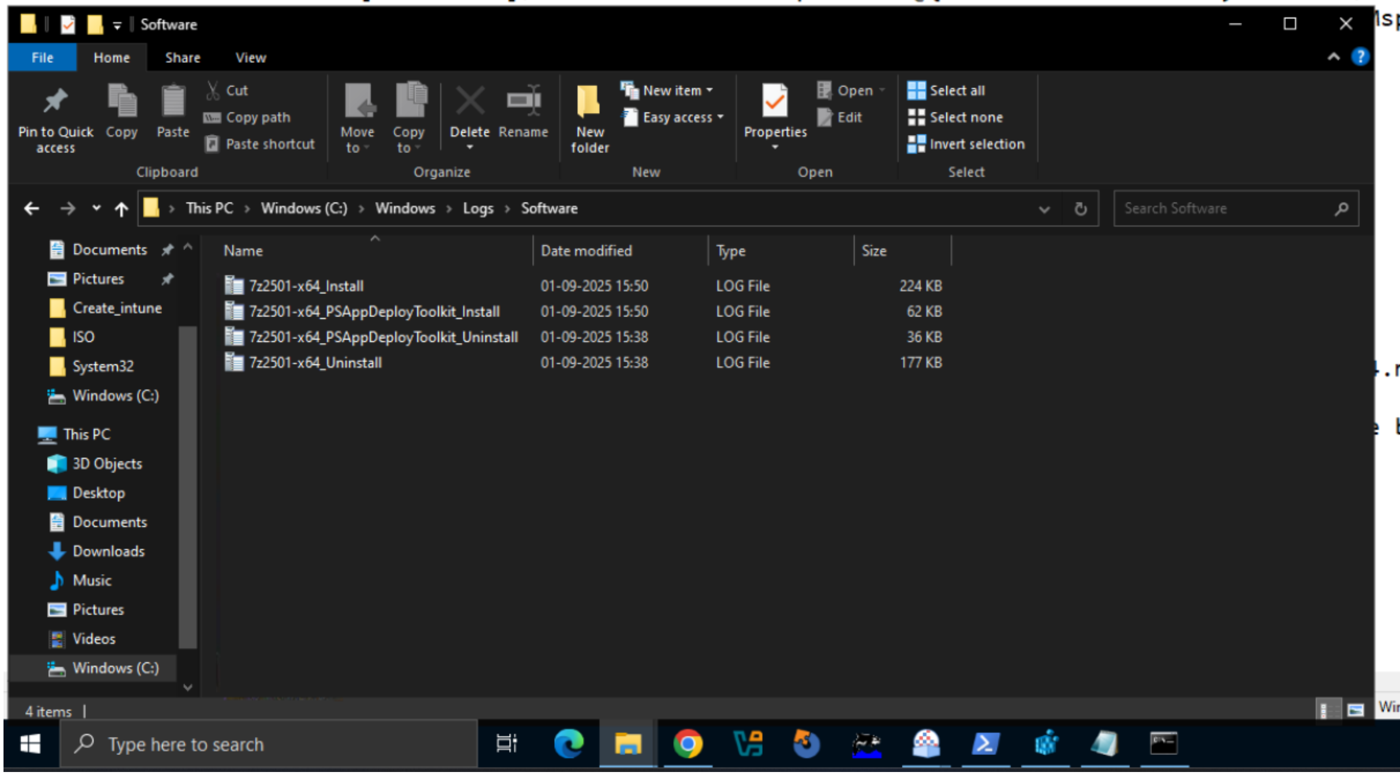
Project_7-zip > Files > Moved

Name	Date modified	Type	Size
7z2501-x64	31-08-2025 20:54	Windows Installer ...	1,950 KB

Verifying results through logs and registry :-

1. **Logs :**

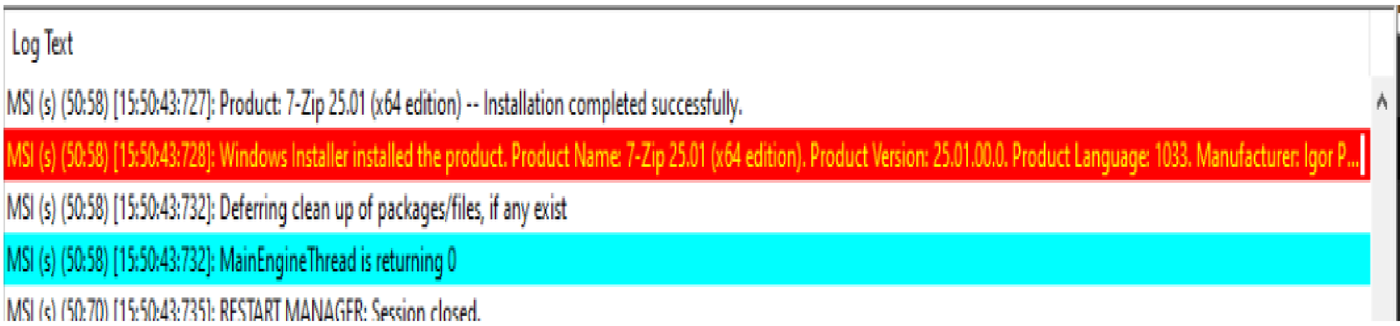
- Check PSADT logs at:
  C:\Windows\Logs\Software or AppDeployToolkit\Logs



This PC > Windows (C:) > Windows > Logs > Software

Name	Date modified	Type	Size
7z2501-x64_Install	01-09-2025 15:50	LOG File	224 KB
7z2501-x64_PSAppDeployToolkit_Install	01-09-2025 15:50	LOG File	62 KB
7z2501-x64_PSAppDeployToolkit_Uninstall	01-09-2025 15:38	LOG File	36 KB
7z2501-x64_Uninstall	01-09-2025 15:38	LOG File	177 KB

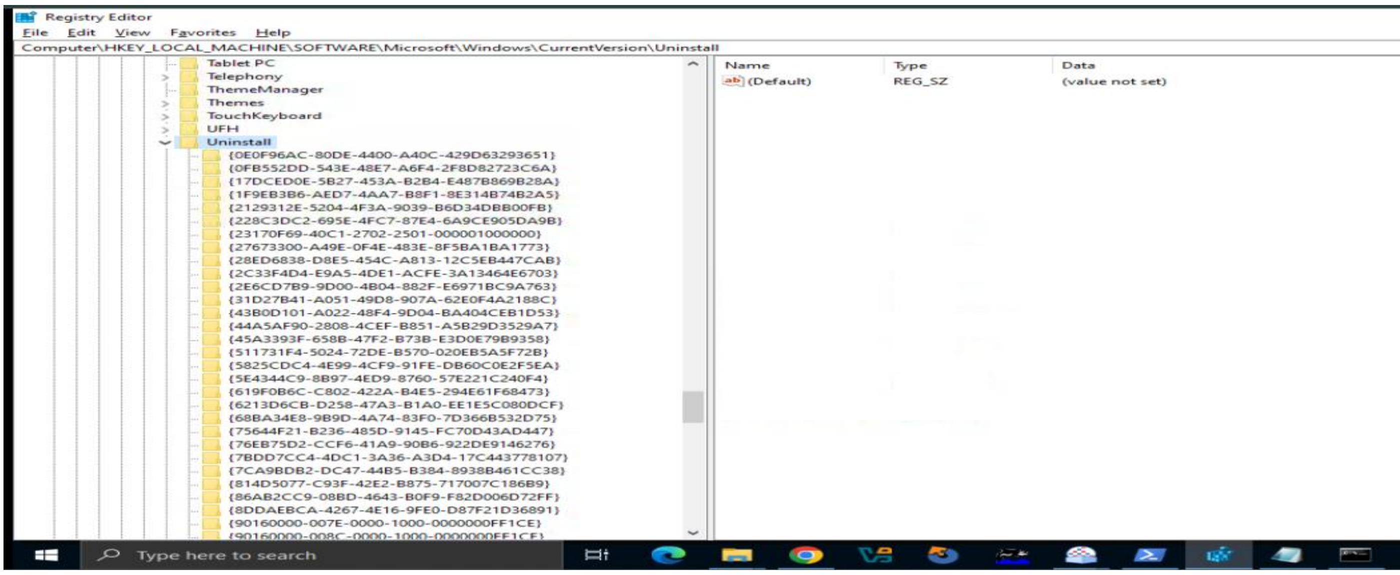
- Confirm installation success message.



```
Log Text
MSI (s) (50:58) [15:50:43:727]: Product: 7-Zip 25.01 (x64 edition) -- Installation completed successfully.
MSI (s) (50:58) [15:50:43:728]: Windows Installer installed the product. Product Name: 7-Zip 25.01 (x64 edition). Product Version: 25.01.00.0. Product Language: 1033. Manufacturer: Igor P...
MSI (s) (50:58) [15:50:43:732]: Deferring clean up of packages/files, if any exist
MSI (s) (50:58) [15:50:43:732]: MainEngineThread is returning 0
MSI (s) (50:70) [15:50:43:735]: RESTART MANAGER: Session closed.
```

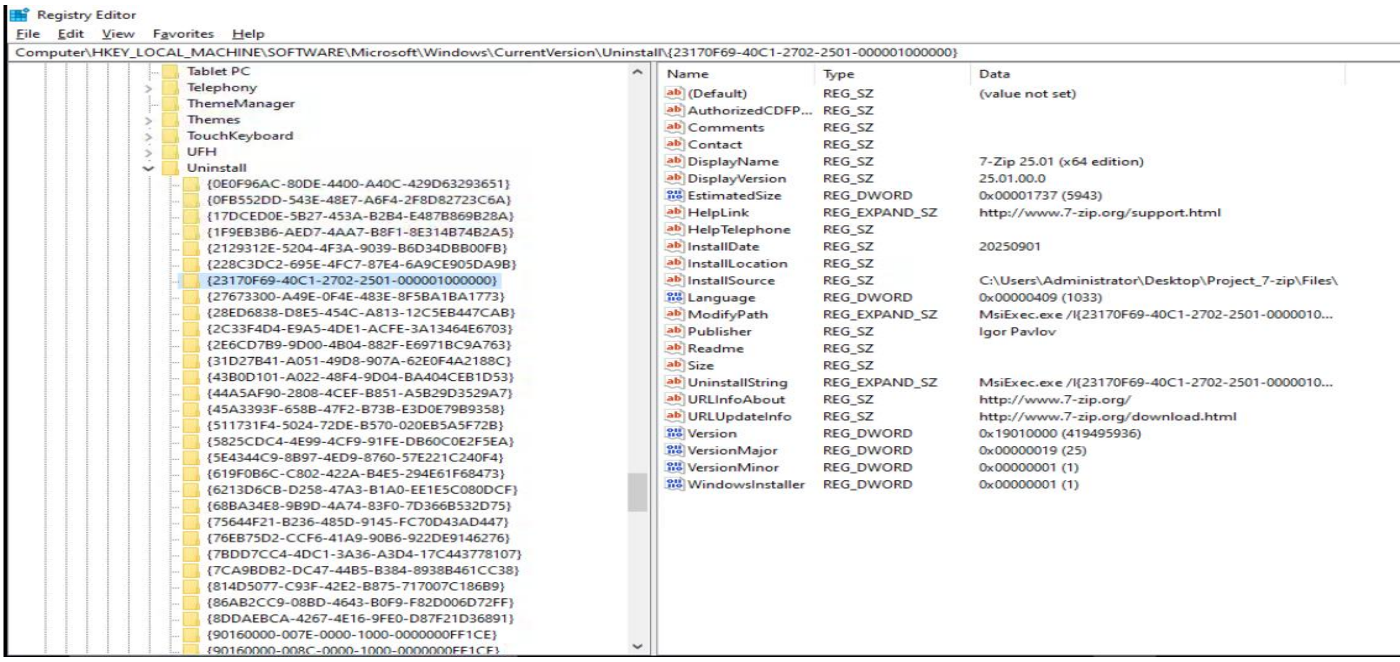
2. **Registry :**

- Open **regedit** → Navigate to:
  HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall



```
Registry Editor
File  Edit  View  Favorites  Help
Computer\HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall
  Tablet PC
> Telephony
  ThemeManager
> Themes
> TouchKeyboard
> UFH
v Uninstall
    {0E0F96AC-80DE-4400-A40C-429D63293651}
    {0FB552DD-543E-48E7-A6F4-2F8D82723C6A}
    {17DCED0E-5B27-453A-B2B4-E487B869B28A}
    {1F9EB3B6-AED7-4AA7-B8F1-8E314B74B2A5}
    {2129312E-5204-4F3A-9039-B6D34DBB00FB}
    {228C3DC2-695E-4FC7-87E4-6A9CE905DA9B}
    {23170F69-40C1-2702-2501-000001000000}
    {27673300-A49E-0F4E-483E-8F5BA1BA1773}
    {28ED6838-D8E5-454C-A813-12C5EB447CAB}
    {2C33F4D4-E9A5-4DE1-ACFE-3A13464E6703}
    {2E6CD7B9-9D00-4B04-882F-E6971BC9A763}
    {31D27B41-A051-49D8-907A-62E0F4A2188C}
    {43B0D101-A022-48F4-9D04-BA404CEB1D53}
    {44A5AF90-2808-4CEF-B851-A5B29D3529A7}
    {45A3393F-658B-47F2-B73B-E3D0E79B9358}
    {511731F4-5024-72DE-B570-020EB5A5F72B}
    {5825CDC4-4E99-4CF9-91FE-DB60C0E2F5EA}
    {5E4344C9-8B97-4ED9-8760-57E221C240F4}
    {619F0B6C-C802-422A-B4E5-294E61F68473}
    {6213D6CB-D258-47A3-B1A0-EE1E5C080DCF}
    {68BA34E8-9B9D-4A74-83F0-7D366B532D75}
    {75644F21-B236-485D-9145-FC70D43AD447}
    {76EB75D2-CCF6-41A9-90B6-922DE9146276}
    {7BDD7CC4-4DC1-3A36-A3D4-17C443778107}
    {7CA9BDB2-DC47-44B5-B384-8938B461CC38}
    {814D5077-C93F-42E2-B875-717007C186B9}
    {86AB2CC9-08BD-4643-B0F9-F82D006D72FF}
    {8DDAEBCA-4267-4E16-9FE0-D87F21D36891}
    {90160000-007E-0000-1000-0000000FF1CE}
    {90160000-008C-0000-1000-0000000FF1CE}

Name        Type     Data
(Default)   REG_SZ   (value not set)

Type here to search
```

- Verify product entry is present (7-Zip or chosen app).



```
Registry Editor
File  Edit  View  Favorites  Help
Computer\HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall\{23170F69-40C1-2702-2501-000001000000}
  Tablet PC
> Telephony
  ThemeManager
> Themes
> TouchKeyboard
> UFH
v Uninstall
    {0E0F96AC-80DE-4400-A40C-429D63293651}
    {0FB552DD-543E-48E7-A6F4-2F8D82723C6A}
    {17DCED0E-5B27-453A-B2B4-E487B869B28A}
    {1F9EB3B6-AED7-4AA7-B8F1-8E314B74B2A5}
    {2129312E-5204-4F3A-9039-B6D34DBB00FB}
    {228C3DC2-695E-4FC7-87E4-6A9CE905DA9B}
    {23170F69-40C1-2702-2501-000001000000}
    {27673300-A49E-0F4E-483E-8F5BA1BA1773}
    {28ED6838-D8E5-454C-A813-12C5EB447CAB}
    {2C33F4D4-E9A5-4DE1-ACFE-3A13464E6703}
    {2E6CD7B9-9D00-4B04-882F-E6971BC9A763}
    {31D27B41-A051-49D8-907A-62E0F4A2188C}
    {43B0D101-A022-48F4-9D04-BA404CEB1D53}
    {44A5AF90-2808-4CEF-B851-A5B29D3529A7}
    {45A3393F-658B-47F2-B73B-E3D0E79B9358}
    {511731F4-5024-72DE-B570-020EB5A5F72B}
    {5825CDC4-4E99-4CF9-91FE-DB60C0E2F5EA}
    {5E4344C9-8B97-4ED9-8760-57E221C240F4}
    {619F0B6C-C802-422A-B4E5-294E61F68473}
    {6213D6CB-D258-47A3-B1A0-EE1E5C080DCF}
    {68BA34E8-9B9D-4A74-83F0-7D366B532D75}
    {75644F21-B236-485D-9145-FC70D43AD447}
    {76EB75D2-CCF6-41A9-90B6-922DE9146276}
    {7BDD7CC4-4DC1-3A36-A3D4-17C443778107}
    {7CA9BDB2-DC47-44B5-B384-8938B461CC38}
    {814D5077-C93F-42E2-B875-717007C186B9}
    {86AB2CC9-08BD-4643-B0F9-F82D006D72FF}
    {8DDAEBCA-4267-4E16-9FE0-D87F21D36891}
    {90160000-007E-0000-1000-0000000FF1CE}
    {90160000-008C-0000-1000-0000000FF1CE}

Name                 Type             Data
(Default)            REG_SZ           (value not set)
AuthorizedCDFP...    REG_SZ
Comments             REG_SZ
Contact              REG_SZ
DisplayName          REG_SZ           7-Zip 25.01 (x64 edition)
DisplayVersion       REG_SZ           25.01.00.0
EstimatedSize        REG_DWORD        0x00001737 (5943)
HelpLink             REG_EXPAND_SZ    http://www.7-zip.org/support.html
HelpTelephone        REG_SZ
InstallDate          REG_SZ           20250901
InstallLocation      REG_SZ
InstallSource        REG_SZ           C:\Users\Administrator\Desktop\Project_7-zip\Files\
Language             REG_DWORD        0x00000409 (1033)
ModifyPath           REG_EXPAND_SZ    MsiExec.exe /I{23170F69-40C1-2702-2501-0000010...
Publisher            REG_SZ           Igor Pavlov
Readme               REG_SZ
Size                 REG_SZ
UninstallString      REG_EXPAND_SZ    MsiExec.exe /I{23170F69-40C1-2702-2501-0000010...
URLInfoAbout         REG_SZ           http://www.7-zip.org/
URLUpdateInfo        REG_SZ           http://www.7-zip.org/download.html
Version              REG_DWORD        0x19010000 (419495936)
VersionMajor         REG_DWORD        0x00000019 (25)
VersionMinor         REG_DWORD        0x00000001 (1)
WindowsInstaller     REG_DWORD        0x00000001 (1)
```

Hence, Installation verification was successful.

Technologies / Tools Used :

- **PowerShell 5.1+:** Automation scripting.
- **PowerShell App Deployment Toolkit (PSADT):** Deployment framework.
- **Target Application Installer (MSI/EXE):** Chosen app for deployment.
- **Virtual Machine (VMware / VirtualBox / Hyper-V):** Testing environment.
- **Windows Event Viewer & Logs:** For analysis.

Tasks Performed :

- Setup of VM environment.
- Downloaded and configured PSADT.
- Structured project folders.
- Customized deployment script for application.
- Executed install / uninstall /modification.
- Validated results through logs and registry.
- Documented complete process with screenshots.

Conclusion :

The project successfully demonstrates how to deploy an application using PSADT. The toolkit makes deployments consistent and reliable while providing logging for troubleshooting. By working in a VM, deployments can be tested safely before moving to production.

References :

- PSADT GitHub
- PSADT Docs – PSADT Usage
- Application vendor documentation (7z2501-x64.msi).
- Community blogs on application packaging.