***Assignment 6***

***Application Deployment Using PSADT***

***Name: Harsh Kumar  
Email: hs150harsh@gmail.com  
Date: August 2025***

***Table of Contents***

1. [*Introduction*](#introduction)
2. [*PSADT Folder Structure*](#psadt-folder-structure)
3. [*Key Components Explained*](#key-components-explained)
4. [*Deploying an MSI Application*](#deploying-an-exe-application)
5. [*Step-by-Step Deployment Process*](#step-by-step-deployment-process)
6. [*Conclusion*](#conclusion)

***Introduction***

*PowerShell App Deployment Toolkit (PSADT) is a powerful framework designed to simplify and standardize application deployment in enterprise environments. This assignment explores its folder structure and demonstrates how to deploy an .exe application using PSADT.*

***PSADT Folder Structure***

*PSADT uses a modular and organized folder hierarchy to streamline deployment workflows. This structure ensures clarity, reusability, and ease of customization.*

*🔹 Folder Overview*

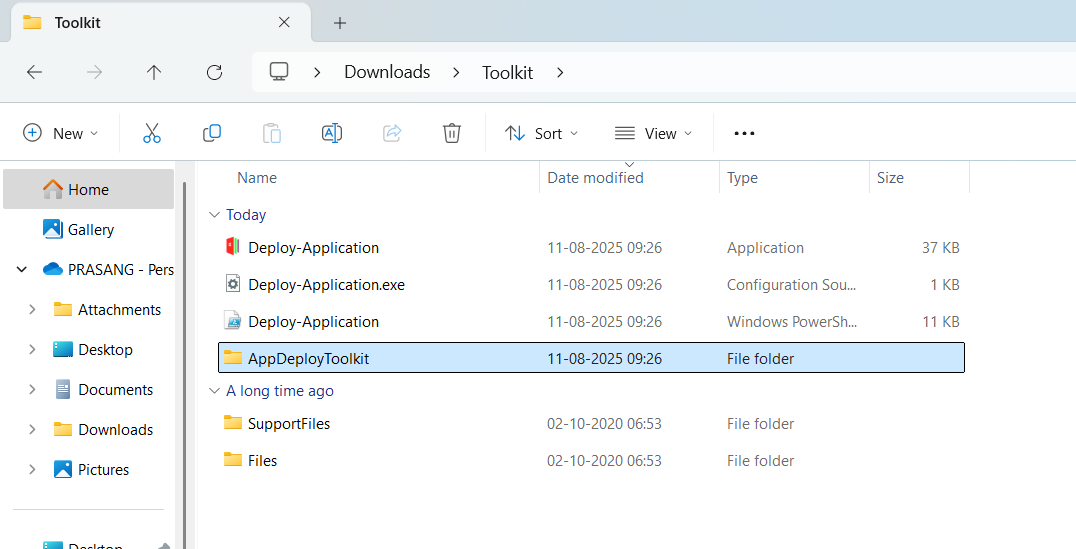
|  |  |
| --- | --- |
| *Folder/File* | *Description* |
| *Toolkit/* | *Core PSADT engine files and functions* |
| *Examples/* | *Sample scripts for reference and reuse* |
| *Files/* | *Location for application installers (.exe, .msi, .appx)* |
| *SupportFiles/* | *Auxiliary files like configs, certificates, and helper scripts* |
| *Deploy-Application.ps1* | *Main PowerShell script controlling the deployment* |
| *Deploy-Application.exe* | *Executable wrapper for running the script with proper execution policy* |
| *AppDeployToolkitConfig.xml* | *Configuration file for toolkit behavior* |
| *AppDeployToolkitMain.ps1* | *Core logic script used by PSADT* |

***Benefits of the Structure***

* *Simplifies deployment management*
* *Supports branding and UI customization*
* *Enables version control integration (e.g., Git)*
* *Enhances modularity and scalability (especially in PSADT v4)*

***Key Components Explained***

* *Deploy-Application.ps1: The heart of the deployment process. You customize this script to define install/uninstall logic.*
* *Deploy-Application.exe: Ensures the script runs with the correct execution policy.*
* *Files/: Keeps installers organized and accessible.*
* *Toolkit/: Contains reusable functions and resources that power the deployment logic.*

**

Deploying an MSI Application

To deploy an .msi installer (e.g., chromestandaloneenterprise64.msi), follow these steps:

Step 1: Place the Installer

Copy your .msi file into the Files folder.  
Example:

.\Files\ googlechromestandaloneenterprise64.msi

A close-up of a computer screen

AI-generated content may be incorrect.

Step-by-Step Deployment Process

🛠️ Step 2: Edit the Deployment Script

Open Deploy-Application.ps1 and locate the installation section. Add the following:

##\*============================[ INSTALLATION ]============================\*##

Show-InstallationWelcome -CloseApps 'Chrome' -AllowDefer

Show-InstallationProgress -StatusMessage 'Installing Chrome, please wait...'

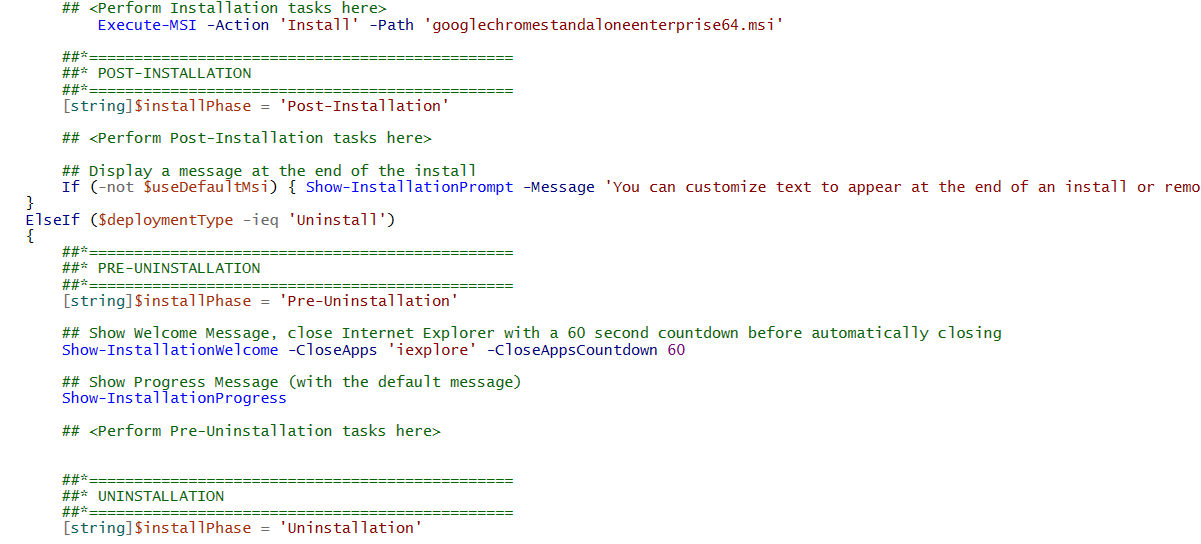
Execute-Process -Path "$dirFiles\ googlechromestandaloneenterprise64.msi " -Parameters "/S"

Show-InstallationPrompt -Message 'Installation complete!' -ButtonRightText 'OK'

For uninstallation:

##\*============================[ UNINSTALLATION ]============================\*##

Execute-Process -Path "$dirFiles\ chromestandaloneenterprise64.msi " -Parameters "/S"



Step 3: Run the Deployment

You can execute the deployment using either method:

* PowerShell Command:

.\Deploy-Application.ps1 -DeploymentType Install

* Executable Wrapper:

Deploy-Application.exe Install

Conclusion

PSADT provides a robust and flexible framework for deploying applications in enterprise environments. By leveraging its structured folder layout and customizable scripting capabilities, IT professionals can ensure consistent, efficient, and error-free deployments.