Name : Manojit Kumar Das

User Id:-34873

Batch:-B5/25VID2550

Email : [dasmonojit026@gmail.com](mailto:dasmonojit026@gmail.com)

**Deployment process using PSADT**

**PSADT Folder Structure and Its Importance**

he **PowerShell App Deployment Toolkit (PSADT)** uses a well-organized folder hierarchy to manage deployment packages and scripts efficiently. This structured approach helps in easy management, customization, and reuse of deployment components, making the application deployment process more streamlined and less error-prone.

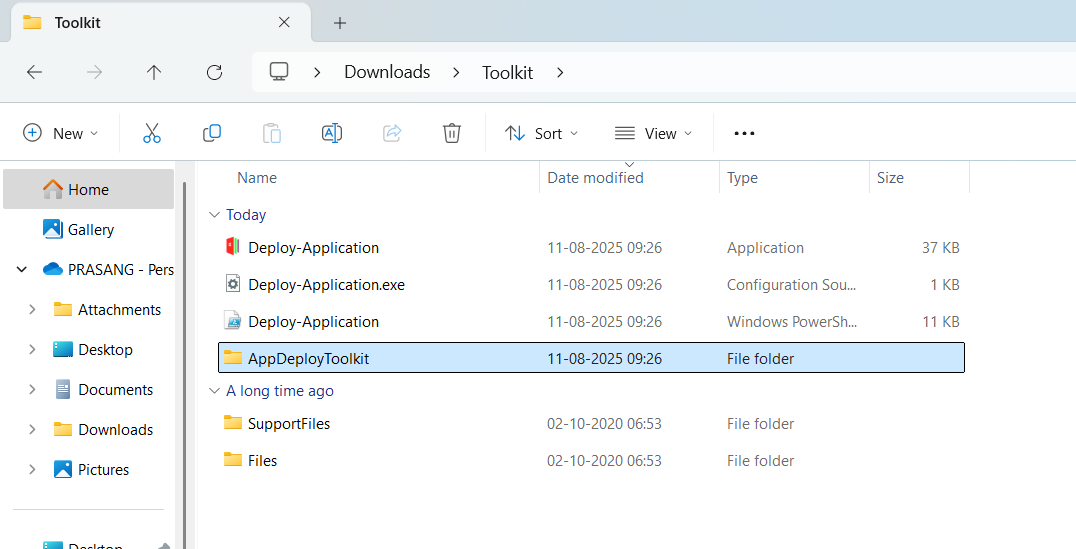
**Folders:**

* **Toolkit/**: Contains core PSADT module files including scripts, manifests, and language resources; essentially the engine of PSADT.
* **Examples/**: Holds example deployment scripts serving as templates for new deployments.
* **Files/**: Stores the application installation files such as .msi, .exe, or .appx.
* **SupportFiles/**: Contains supporting files like configuration files, certificates, or helper scripts required during deployment.
* **Deploy-Application.ps1**: The main PowerShell script orchestrating the deployment process, customized per application.
* **Deploy-Application.exe**: A wrapper to run the PowerShell script with the correct execution policy.

**Additional Highlights:**

* PSADT supports customization for branding (banners, icons), UI, and scripts.
* Version control (e.g., Git) is recommended to track changes and maintain deployment scripts.
* The modular and flexible structure introduced in PSADT v4 makes managing and customizing deployments easier.

Understanding and using this folder structure effectively allows IT professionals to create, deploy, and manage application deployments with greater ease and consistency.



**psadt .exe application deployment process**

When you download PSADT, you’ll get a folder with:

Deploy-Application.ps1 ← Main script you edit

Deploy-Application.exe ← Wrapper to run the .ps1

Toolkit\ ← PSADT core functions

Files\ ← Place for your application installers

AppDeployToolkitConfig.xml

AppDeployToolkitMain.ps1

For deploying an .exe app, you’ll mainly work with:

* \**Files\** → put your .exe installer here.
* **Deploy-Application.ps1** → edit this to install/uninstall.

**Place Your .exe Installer**

Example:

CopyEdit

.\Files\7z2501-x64.exe

This keeps things organized and lets PSADT find the installer easily.

**Edit Deploy-Application.ps1**

Inside the script, find the **##\*============================[ INSTALLATION ]============================\*##** section and add:

powershell

CopyEdit

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

# Install 7-Zip silently

Show-InstallationWelcome -CloseApps '7zFM' -AllowDefer

Show-InstallationProgress -StatusMessage 'Installing 7-Zip, please wait...'

Execute-Process -Path "$dirFiles\7z2501-x64.exe" -Parameters "/S"

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

For uninstall, in the **##\*============================[ UNINSTALLATION ]============================\*##** section:

powershell

CopyEdit

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

# Example uninstall (if your exe supports /uninstall or /S)

Execute-Process -Path "$dirFiles\7z2501-x64.exe" -Parameters "/S" # or actual uninstall command

**Run the Deployment**

You have **two ways**:

* **PowerShell directly:**

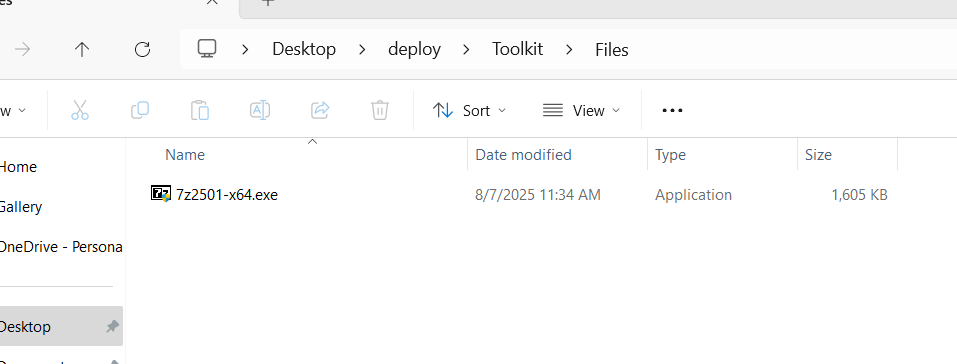
powershell

CopyEdit

.\Deploy-Application.ps1 -DeploymentType Install







**OVERVIEW OF TOOL**

**What is PSADT?**

The PowerShell App Deployment Toolkit (PSADT) is a framework for streamlining and standardizing application deployments in enterprise environments. It provides a collection of pre-built PowerShell functions and a structured approach to handle various aspects of application installation, uninstallation, and updates. PSADT simplifies the complexities of application packaging by offering a consistent and reliable deployment experience.

**Features:-**

1. Standardized Deployment Process: PSADT provides a consistent framework for deploying applications, making the process more reliable and easier to troubleshoot.
2. Pre- and Post-Installation Actions: It allows for performing actions before and after the main installation process, such as closing specific applications or running custom scripts.
3. User Interaction: PSADT can display user-friendly prompts during installation, allowing for tasks like closing in-use applications or providing options for deferring the installation.
4. Customizable: It allows for customization of the user interface, including the banner, language support, and the addition of custom functions.
5. Open-Source: PSADT is an open-source project, encouraging community contributions and allowing for transparency and flexibility.
6. Supports various installers: PSADT can handle different types of installers, including MSI, EXE, and more complex setups with multiple components.
7. Integration with other tools: PSADT can be used to deploy applications to various platforms, including Microsoft Endpoint Configuration Manager (MECM) and Microsoft Intune.

**Working Principle:-**

1. Packaging: The application is packaged with the PSADT files, including the Deploy-Application.ps1 script, which is the core of the deployment process.
2. Customization: The Deploy-Application.ps1 script is customized with the specific installation steps, pre- and post-installation actions, and user interface elements.
3. Execution: The deployment is initiated by running the Deploy-Application.exe or Deploy-Application.ps1 file.
4. Installation: PSADT handles the installation process based on the instructions in the Deploy-Application.ps1 script, including any pre- or post-installation tasks and user interaction.

**FOLDER STRUCTURE**

The PSADT uses a structured folder hierarchy to organize deployment packages and scripts.The structure allows for easy management and customization of deployments.

**Main Folders:-**

1. **Toolkit/**: This folder contains the core PSADT module files, including the manifest, script, and language strings.
2. **Examples/**: This folder holds example deployment scripts for various applications, serving as templates for new deployments.
3. **Files/**: This is where you place the application's installation files, such as .msi, .exe, or .appx files.
4. **SupportFiles/**: This folder is used to store supporting files required for the deployment, like configuration files, scripts, or certificates.
5. **Deploy-Application.ps1**: This is the main PowerShell script that orchestrates the deployment process. It's where you customize the script for your specific application.
6. **Deploy-Application.exe**: This is a wrapper for the Deploy-Application.ps1 script. It ensures that the correct PowerShell execution policy is set.

**Main Consideration:-**

1. **Modular Structure:** PSADT v4 introduces a more modular structure with dedicated folders for different types of files and settings, making it easier to manage and customize.
2. **File and Folder Structure:** The deployment structure is designed to be flexible and user-friendly, allowing for customization and reuse.
3. **Customization:** You can customize the PSADT's UI, branding, and deployment scripts to suit your specific needs.

**TOOL CONFIGURATION IN PS APP DEPLOY TOOLKIT**

**Overview:-** The PowerShell App Deployment Toolkit (PSAppDeployToolkit) is a scripting framework that simplifies application deployment. It allows for customization and control over the deployment process, including pre-installation checks, UI customization, and logging options. To configure PSAppDeployToolkit, you modify the AppDeployToolkitConfig.xml file, which allows you to set default log locations, message icons, banners, logging options, and more.

**Break Down of the Process:-**

1. **Download and Extract:** Download the PSAppDeployToolkit and extract the zip file.
2. **Locate Configuration File:** Navigate to the Toolkit\AppDeployToolkit directory and find the AppDeployToolkitConfig.xml file.
3. **Edit Configuration:** Open the AppDeployToolkitConfig.xml file and customize it according to your needs.
4. **Toolkit Options:** Configure general toolkit settings like required administrative rights, temporary file paths, and logging options.
5. **Banner, Logo & Icon Options:** Customize the UI banner, logos, and icons.
6. **MSI Options:** Configure logging options, log paths, and installation parameters for MSI installations.
7. **UI Options:** Set general UI options like balloon notifications, script timeouts, and exit codes.
8. **UI Messages:** Edit or add languages for the user interface.

**Benefits of Using It:**-

It has several benefits.

Like:Centralized Configuration, Consistency, Customization, Simplified Scripting

**TEMPLATE SCRIPT IN PS APP DEPLOY TOOLKIT**

**What is Template Script?**

The PowerShell Application Deployment Toolkit (PSAppDeployToolkit) provides template scripts for application deployment. These scripts, like Deploy-Application.ps1, are designed to streamline the deployment process and ensure consistency across deployments.

**There are 5 most important topic about the Template Script, these are:-**

* **What is PSAppDeployToolkit (PSADT)?**
* A framework that makes PowerShell app deployment easier for IT admins.
* Comes with built-in functions and UI pop-ups for common install/uninstall tasks.
* Includes a ready-to-edit template script: Deploy-Application.ps1**.**
* **Using Deploy-Application.ps1**
* Template script for installing or uninstalling apps.
* Has three install sections: Pre-Install, Install, Post-Install.
* Relies on AppDeployToolkitMain.ps1 for core logic.
* Open and edit it in PowerShell ISE to customize for your app.
* **Creating a New Deployment**
* Use built-in commands to make new projects:

For PSADT v3:

New-ADTTemplate -Destination C:\Temp\MyAppDeployment -Name "MyOldAppDeployment" -Version 3

For PSADT v4:

New-ADTTemplate -Destination C:\Temp\MyAppDeployment -Name "MyAppDeployment"

* **Getting Templates**
* Download from [PSADT GitHub Releases](https://github.com/PSAppDeployToolkit/PSAppDeployToolkit/releases).
* Files: PSAppDeployToolkit\_Template\_v3.zip or PSAppDeployToolkit\_Template\_v4.zip.
* **Customizing the Script**
* Edit parameters, add commands, and tweak the UI to suit your deployment needs.
* Acts as a base for both simple and complex installs.

**COMMANDS FOR INSTALLATION/UNINSTALLATION IN psappdeploy kit**

In the PowerShell App Deployment Toolkit (PSAppDeployToolkit), the main commands for installation and uninstallation are Execute-MSI and Uninstall-ADTApplication respectively. Execute-MSI is used to run MSI installers, while Uninstall-ADTApplication handles uninstalling applications**.**

1. **Installation:-**

* **Execute-MSI** → Runs MSI installers.  
  Example:

Execute-MSI -Action Install -Path "C:\path\to\app.msi" -DeployMode Silent

1. **Uninstallation:-**

* **Uninstall-ADTApplication** → Removes apps installed with PSADT.
* **Execute-MSI -Action Uninstall** → Removes MSI apps.  
  Example:

Execute-MSI -Action Uninstall -Path "{Product-GUID}"

1. **Other Useful Commands:-**

* **Get-RunningProcesses** → Check running apps before install/uninstall.
* **Show-WelcomePrompt** → Display a start message to users.
* **Remove-MSIApplications** → Remove specific MSI apps.
* **Unblock-ADTAppExecution** → Allow a blocked app to run.
* **Unregister-ADTDll** → Unregister a DLL file.