# Batch Information:

* **Batch Start Date:** 2025-08-04
* **Batch Name:** WiproNGA\_DWS\_B5\_25VID2550
* **First Name:** Aayush
* **Last Name:** Kumar
* **User ID:** 34758
* **Batch ID:** 25VID2550
* **Email ID:** theaayushtagore@gmail.com

**Assignment**

**The Importance of the PowerShell App Deployment Toolkit (PSADT)**

The PowerShell App Deployment Toolkit (PSADT) is a powerful framework created to make the process of deploying applications easier, more organized, and more reliable. It is built using PowerShell, and its main purpose is to provide a ready-made, consistent structure for software deployment.

When you deploy applications in large environments (for example, in companies with hundreds of computers), the process can easily get messy if every deployment is done differently. PSADT solves this problem by using:

* A clear folder structure that separates core toolkit files, application installers, support files, and example scripts.
* Pre-built functions that handle common deployment tasks such as installing, uninstalling, showing progress bars, prompting users, and logging.
* Customization options to adapt the toolkit to your organization’s style and requirements.

By following this standardized approach, PSADT helps you:

1. Work faster – You don’t have to start from scratch every time.
2. Avoid mistakes – The structure makes it harder to mix up files or run the wrong commands.
3. Maintain deployments easily – If you need to update, fix, or re-run a deployment, everything is already organized and documented.
4. Reuse work – You can take an old project, adjust it slightly, and redeploy without starting over.

**Folder Structure of PSADT**

When you download PSADT, you’ll notice a specific folder layout. Each folder has a purpose, and sticking to this layout makes life much easier:

1. Toolkit/
   * This is the heart of PSADT.
   * Contains the core scripts, PowerShell module files, manifests, and language resources.
   * Without this folder, the toolkit will not work.
2. Examples/
   * Contains sample deployment scripts.
   * You can use these as templates or reference material when creating your own deployments.
3. Files/
   * This is where you put your application installer files (e.g., .exe, .msi, .appx).
   * Keeping all installers here means PSADT can easily find them without needing complex paths.
4. SupportFiles/
   * Stores extra resources needed during deployment, like:
     + Configuration files
     + Security certificates
     + Helper scripts
   * Anything that supports the main installer but is not the installer itself goes here.
5. Deploy-Application.ps1
   * This is your main script.
   * You edit this file to control what happens during the deployment (install, uninstall, pre-install checks, post-install cleanup).
   * It’s divided into sections for clarity:
     + Pre-installation tasks
     + Installation tasks
     + Post-installation tasks
6. Deploy-Application.exe
   * A wrapper executable that launches the .ps1 script.
   * Ensures the PowerShell execution policy is set correctly so your script runs smoothly.

**Extra Features of PSADT**

* Branding Customization – You can change banners, logos, and user interface elements to match your company’s style.
* Version Control – Using tools like Git is highly recommended so you can track changes, roll back if needed, and maintain a history of all updates.
* Modular Framework (from v4) – The newer versions are designed to be even more flexible, letting you easily adapt scripts to different applications.

**What You Get When You Download PSADT**

The downloaded package usually contains:

* Deploy-Application.ps1 → The main script you will customize.
* Deploy-Application.exe → The launcher for the .ps1 script.
* Toolkit/ → Core functions and framework files.
* Files/ → Folder where you store application installers.
* AppDeployToolkitConfig.xml → Configuration file for the toolkit.
* AppDeployToolkitMain.ps1 → The main PowerShell script that supports all deployment operations.

**Deploying an Application with an .exe Installer (Example: Firefox)**

When deploying a program that uses an .exe installer, you mainly work with:

1. Files folder – Place your .exe here.  
   Example path:

makefile

CopyEdit

C:\Users\thaku\Downloads\PSAppDeployToolkit\_v3.8.4\Toolkit\Files

1. Deploy-Application.ps1 – Add your installation logic here.

**Editing Deploy-Application.ps1 for Firefox**

Inside Deploy-Application.ps1, find the section:

shell

CopyEdit

##============================[ INSTALLATION ]============================##

Add this code:

powershell

CopyEdit

# Install Firefox silently

Show-InstallationWelcome -CloseApps '' -AllowDefer

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

Execute-Process -Path "$dirFiles\firefox.exe" -Parameters "/S"

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

For uninstallation, go to:

shell

CopyEdit

##============================[ UNINSTALLATION ]============================##

And add:

powershell

CopyEdit

Execute-Process -Path "$dirFiles\firefox.exe" -Parameters "/S"

*(Only works if the Firefox installer supports silent uninstall with /S.)*

Running Deployments in SYSTEM Context

* Some applications require SYSTEM-level permissions to install for all users.
* SYSTEM context gives full permissions, which is important for enterprise deployments.
* You can test your script locally in SYSTEM context using PsExec before rolling it out.

**PSAppDeployToolkit Overview**

* PSADT is essentially a collection of ready-made PowerShell functions for deployment.
* It provides a standard script template with three phases:
  1. Pre-install – Prepare the system (close apps, check requirements).
  2. Install – Install the software.
  3. Post-install – Cleanup and final confirmations.
* The main script (Deploy-Application.ps1) works closely with AppDeployToolkitMain.ps1.

**Creating a Deployment Template**

You can create a fresh deployment template using:

* For version 3:

powershell

CopyEdit

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

* For version 4:

powershell

CopyEdit

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

You can also download templates from GitHub releases and customize them.

**Toolkit Configuration (AppDeployToolkitConfig.xml)**

The configuration file allows you to control global settings for all deployments.

Steps:

1. Download and extract PSADT.
2. Open AppDeployToolkitConfig.xml.
3. Adjust settings for:
   * Admin rights
   * Temporary paths
   * Log folder
   * UI branding (banners, icons)
   * MSI defaults (parameters, log location)
   * Timeouts, notifications, exit codes
4. Save and apply – changes affect all deployments.

Benefits:

* One file controls everything.
* Consistent behavior across all applications.
* Saves time by avoiding repetitive edits.

**MSI/MSP Logging in PSADT**

PSADT includes commands like Execute-MSI and Execute-MSP for handling .msi installers and .msp patches with automatic logging and error handling.

Logging Parameters:

* -LogName → Name of the log file.
* -LogPath → Where logs are stored.
* -LogVerbosity → How detailed the logs are (Verbose, Informational, Error).

Examples:

* Install an MSI:

powershell

CopyEdit

Execute-MSI -Action Install -Path "C:\MyApps\MyPackage.msi" -LogName "InstallLog" -LogVerbosity Verbose

* Patch an MSP:

powershell

CopyEdit

Execute-MSI -Action Patch -Path "C:\MyApps\Patch.msp" -LogName "PatchLog"

* Uninstall MSI:

powershell

CopyEdit

Execute-MSI -Action Uninstall -Path "{PRODUCT-CODE}" -LogName "UninstallLog"