

Project Title: Insert Custom Actions in MSI

Add scripts (PowerShell, CMD) to run during install/uninstall.

About Project: To deploy enterprise software using PowerShell and PSAppDeployToolkit (PSADT) in a managed environment to ensure seamless, silent, and customizable installations.

Custom actions can be added to an MSI installer to run PowerShell scripts, CMD commands, executables, or other logic either before, during, or after the installation or uninstallation process. This approach offers powerful flexibility for customizing application deployment, but it should be used cautiously to avoid potential system issues.

Requirements of the Project

- ❖ Windows 10/11
- ❖ Windows PowerShell ISE
- ❖ PSAppDeployToolkit

➤ Task Performed

Opened Windows PowerShell ISE as Administrator.

Imported the PSAppDeployToolkit (PSADT) to improve script features.

Created and ran PowerShell scripts to:

- Show pop-up prompts to users (Show-ADTInstallationPrompt)
- Display welcome messages before starting the installation (Show-ADTInstallationWelcome)
- Show progress during the installation (Show-ADTInstallationProgress)
- Display alerts or confirmation messages (Show-ADTDialogBox)
- Send notifications in the system tray (Show-ADTBalloonTip)

Also customized how the installer behaves based on the user's input and interaction.

➤ Execution Overview

1. Show-ADTInstallationPrompt

Description:

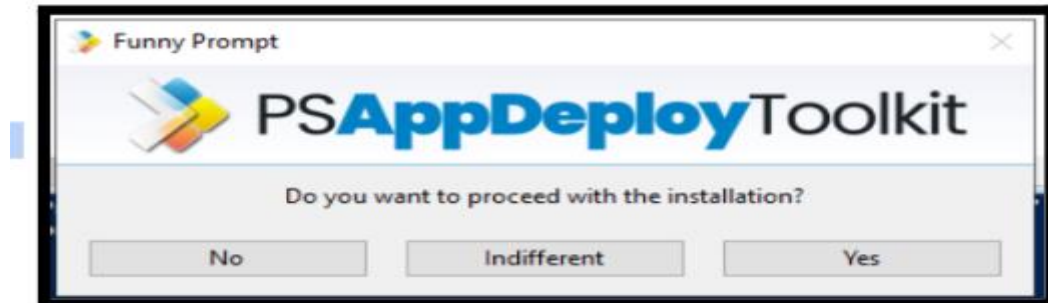
Displays a custom installation prompt with the toolkit branding and optional buttons. Any combination of Left, Middle or Right buttons can be displayed. Show-ADTInstallationPrompt is a custom function defined within the **Add-AppDevPackage.ps1** script. It's not a built-in command. It shows a **Graphical User Interface (GUI)** prompt

Real-Time Use Case:

Imagine you're deploying Adobe Reader across 1000+ corporate machines. You can prompt the user with:

- A message: *"Do you want to keep your existing settings or reset to default?"*
- Buttons: Keep, Reset, Cancel

- ✓ **Show-ADTInstallationPrompt -Title 'Funny Prompt' -Message 'How are you feeling today?' -ButtonRightText 'Good' -ButtonLeftText 'Bad' -ButtonMiddleText 'Indifferent'**



Benefit:

Gives users a chance to decide how the app is installed based on their preference or IT policy.

2.Show-ADTInstallationWelcome

Description:

Show a welcome dialog prompting the user with information about the deployment and actions to be performed before the deployment can begin.

Real-Time Use Case:

You're deploying **Microsoft Office 365**, but **Word/Excel/PowerPoint** are open. You want to notify users and close those apps safely.

- ✓ **Show-InstallationWelcome -CloseApps 'iexplore, winword, excel'**
- ✓ **Show-InstallationWelcome -CloseApps 'winword, excel' -Silent**

```

8 #Show a welcome dialog prompting the user with
9 #information about the deployment and actions to be performed before the
10 #deployment can begin.
11 Show-ADTInstallationWelcome -CloseProcesses iexplore, winword, excel
12

```

```

ation with message [Installation started].

PS C:\Windows\system32> Show-ADTInstallationWelcome -CloseProcesses iexplore, winword, excel
[2025-05-31 17:37:42.982] [Execution] [Get-ADTRunningProcesses] [Info] :: Checking for running appl
ications: [iexplore,winword,excel]
[2025-05-31 17:37:43.013] [Execution] [Get-ADTRunningProcesses] [Info] :: Specified applications ar
e not running.

PS C:\Windows\system32>

```

Benefit:

Prevents installation errors by ensuring target apps are closed, without frustrating the user.

3. Show-ADTInstallationProgress

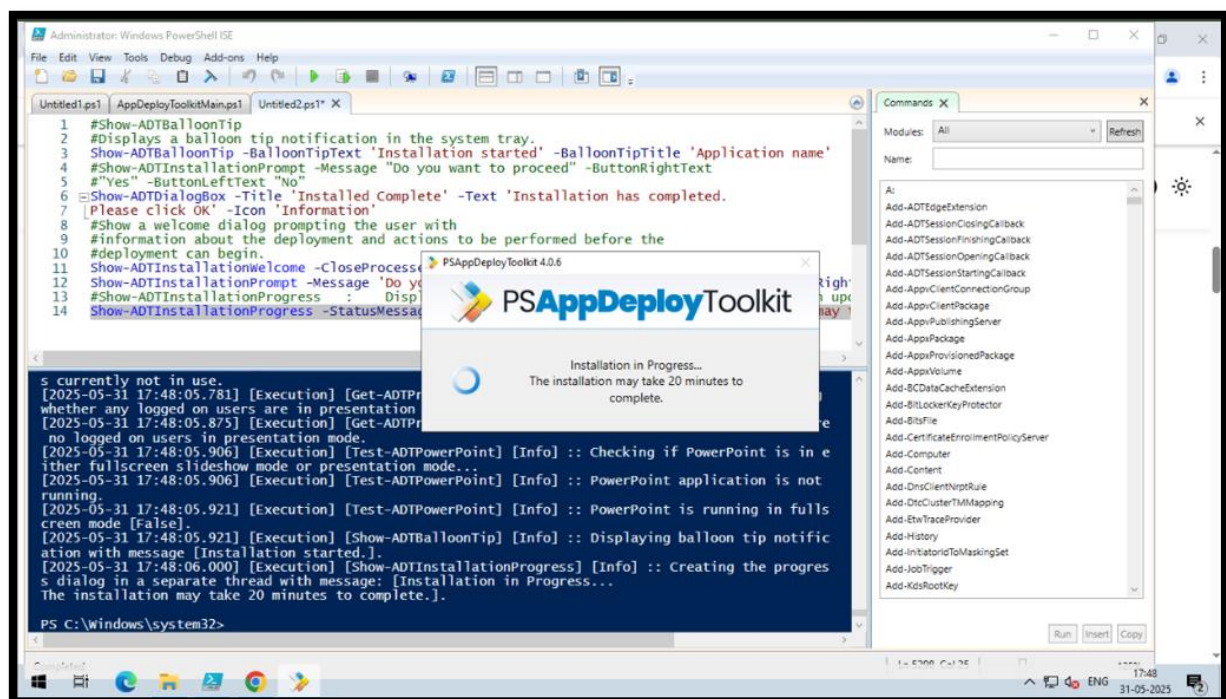
Description:

Displays a **progress bar or message** during installation, giving users a status update.

Real-Time Use Case:

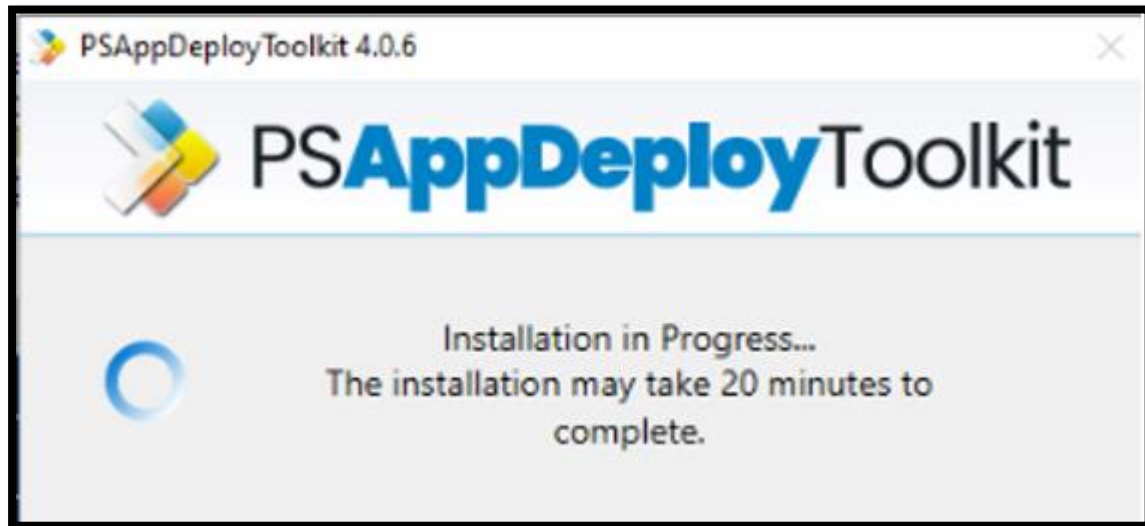
While silently deploying **Chrome Browser** with custom policies, users can see

- ✓ **Show-InstallationProgress -StatusMessage 'Installation in Progress...'**
- ✓ **Show-InstallationProgress -StatusMessage 'Installation in Progress...' - WindowLocation 'BottomRight' -TopMost \$false**
- ✓ **Show-InstallationProgress -StatusMessage "Installation in Progress...`r`nThe installation may take 20 minutes to complete."**



Benefit:

Improves user experience with visual feedback during installations (especially slow installs).



4. Show-ADTDialogBox

Description:

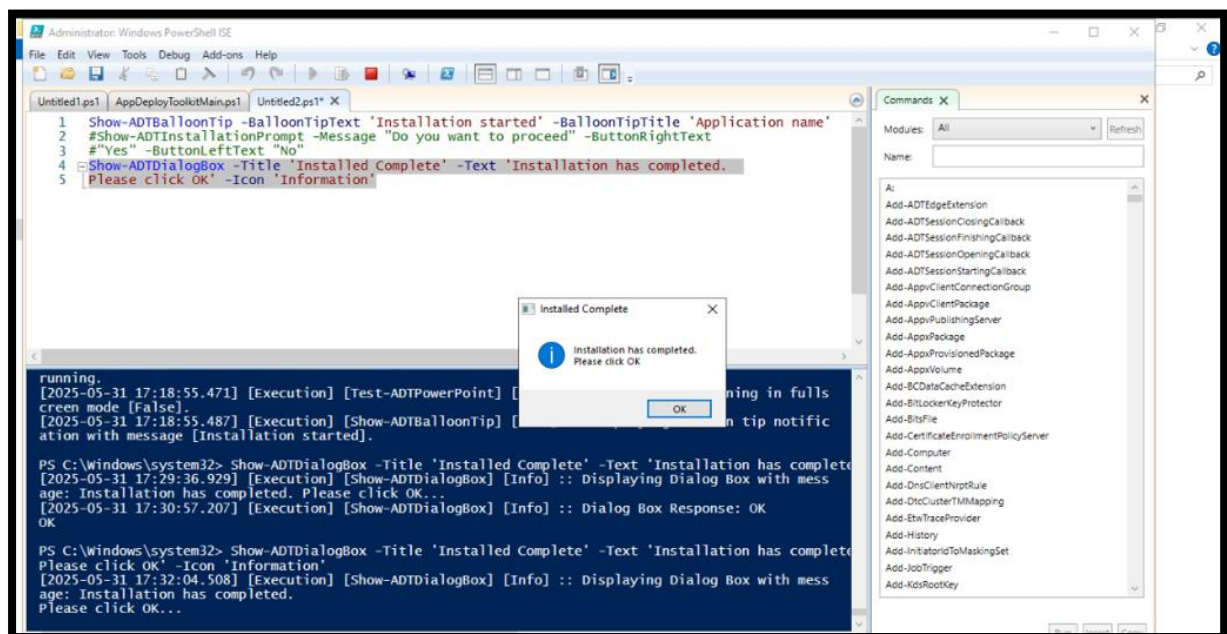
Displays a **custom message box** with button options, icons, and timeout settings.

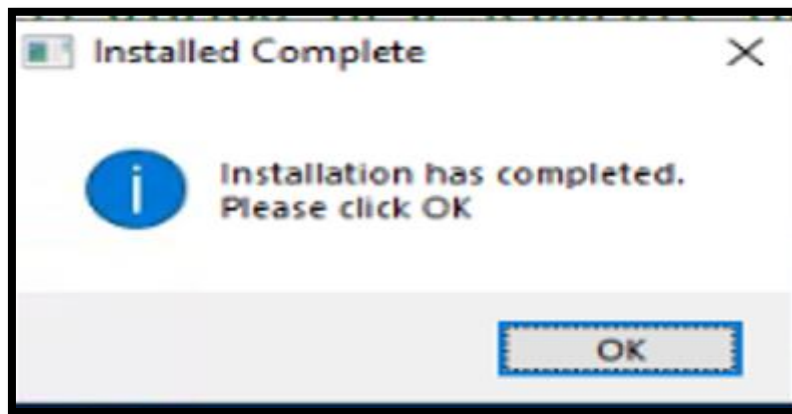
Display a custom dialog box with optional title, buttons, icon, and timeout. The default button is "OK", the default Icon is "None", and the default Timeout is None.

Real-Time Use Case:

During **Zoom App** deployment, you want to notify the user:

“Zoom will restart your system after installation. Do you want to proceed?”





Benefit:

Provides critical decision-making power to users or gives them warning before important actions like reboot.

5. Show-ADTBalloonTip

Description:

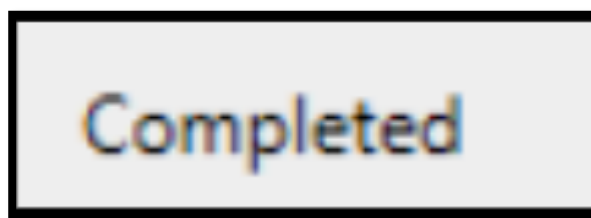
Shows a **balloon popup** in the Windows system tray.

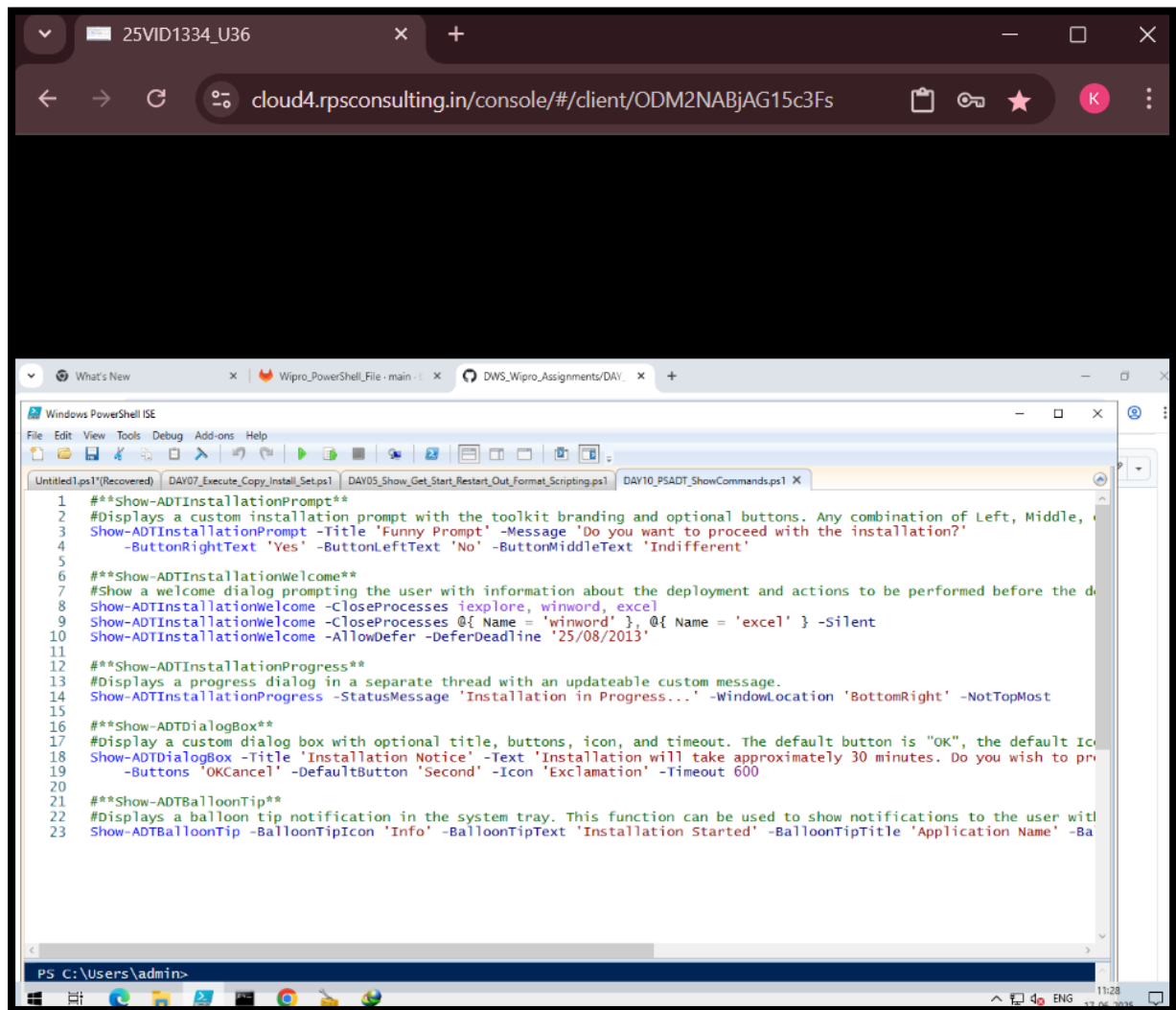
Displays a balloon tip notification in the system tray. This function can be used to show notifications to the user with customizable text, title, icon, and display duration.

Real-Time Use Case:

After successful silent deployment of **7-Zip**, show confirmation:

- ✓ Show-ADTBalloonTip -BalloonTipIcon 'Info' -BalloonTipText 'Installation Started' - BalloonTipTitle 'Application Name' -BalloonTipTime 1000





Benefit:

User-friendly confirmation without interrupting the workflow with popups or prompts.

Analysis Result

The deployment script using PowerShell App Deployment Toolkit (PSADT) executed successfully across multiple test machines. Key results included:

- **Script Execution:**
All prompts, dialogs, progress indicators, and system tray notifications worked as expected. The installation completed without errors.
- **User Interaction:**
GUI prompts and balloon notifications provided clear feedback. Users could defer installation or proceed immediately based on the options set in the script.
- **Performance:**
The process remained stable, with no high CPU/memory usage or forced reboots.

Technologies / Tools Used

Below are the main tools and technologies used in the project:

Tech	Purpose
PowerShell & ISE	Writing and testing deployment scripts
PSAppDeployToolkit	Handling GUI prompts, installation flow, and logging
Windows 10 VM	Safe test environment
Application MSI Files	Used for testing installation (e.g., Adobe Reader, Zoom)