Insert Custom Actions in MSI

❖ About the Project:

- The PSADT is a PowerShell-based framework that simplifies application deployment for enterprises.
- Adding **custom actions to an MSI installer** allows us to run **PowerShell scripts, CMD commands, executables** or other logic **before, during or after** the installation/uninstallation process.
- This is a powerful way to customize app deployment but it must be used carefully to avoid system issues.
- This is particularly useful for pre/post-install tasks like **setting registry keys**, **checking prerequisites** or **cleaning up files**.

❖ Project Requirements:

The basic requirements for executing these commands are Windows 10/11, Windows PowerShell ISE, PSAppDeployToolkit.

❖ Task Performed:

- Performed PSADT commands for Installation / Uninstallation.
- Wrote scripts in PowerShell and executed them.
- Also, researched on real-life uses of these commands.
- Displayed alerts and dialog boxes for confirmations.
- Displayed interactive prompts and presented welcome messages before installation.
- Displayed real-time progress indicators.

❖ Execution Overview

Commands:

1) **Show-ADTInstallationPrompt**

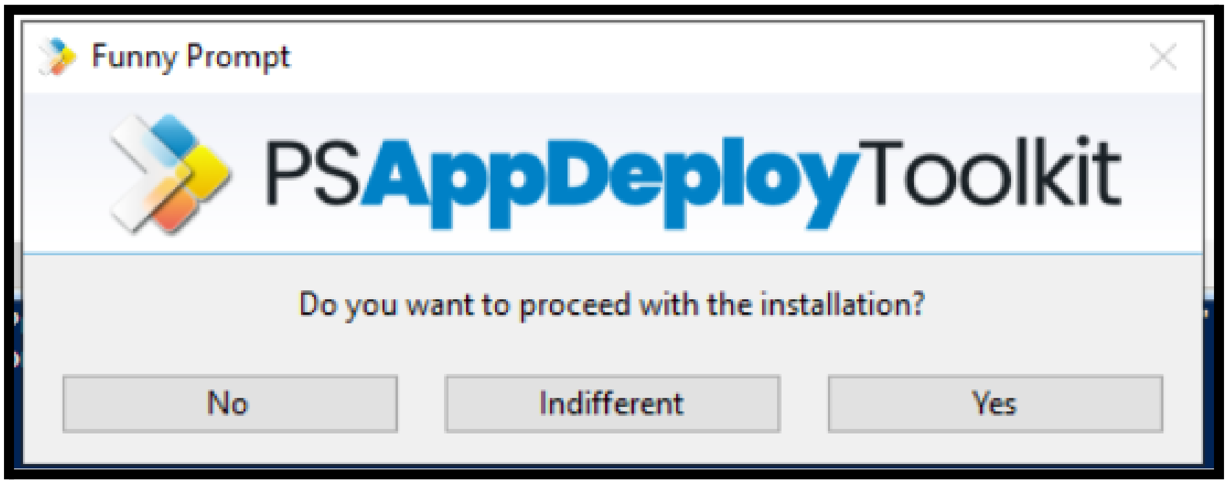
   Displays a custom installation prompt with the toolkit branding and optional buttons. Any combination of Left, Middle or Right buttons can be displayed.

   Show-ADTInstallationPrompt is a custom function defined within the **Add-AppDevPackage.ps1** script. It's not a built-in command.

   Uses: It shows a **Graphical User Interface (GUI)** prompt to guide the user through: i) Installing the app ii) Installing the certificate iii) Handling errors like unsigned packages or dependency issues

   Shows a friendly install window via Show-ADTInstallationPrompt and automates the whole sideloading process.

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


Funny Prompt

Do you want to proceed with the installation?

2) Show-ADTInstallationWelcome

It's a helper PowerShell function defined inside **Add-AppDevPackage.ps1**.
It’s not a built-in PowerShell cmdlet as it’s written by Microsoft to provide a **graphical welcome screen** when a user runs the installation script.

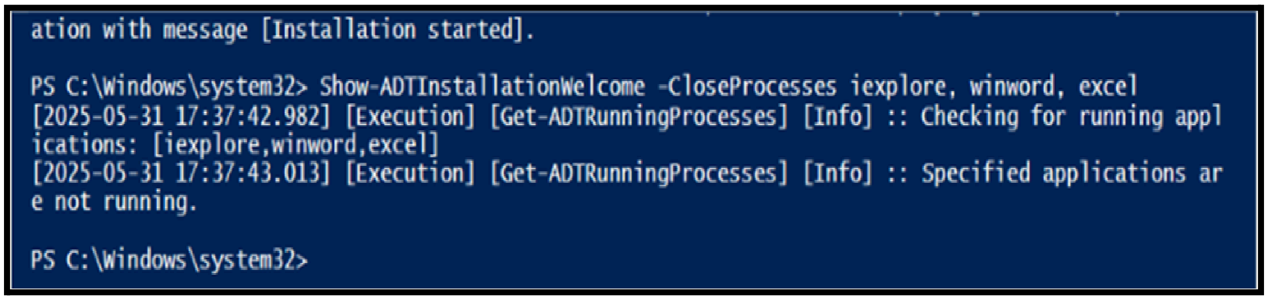
Uses: It displays a graphical welcome dialog.
It explains what’s about to happen.
It improves usability for non-technical users
It may show information like the app name, publisher, version, etc.

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

Commands: i) Show-ADTInstallationWelcome -CloseProcesses iexplore, winword, excel

ii) Show-ADTInstallationWelcome -CloseProcesses @{ Name = 'winword' }, @{ Name = 'excel' } -Silent

iii) Show-ADTInstallationWelcome -AllowDefer -DeferDeadline '25/08/2013'



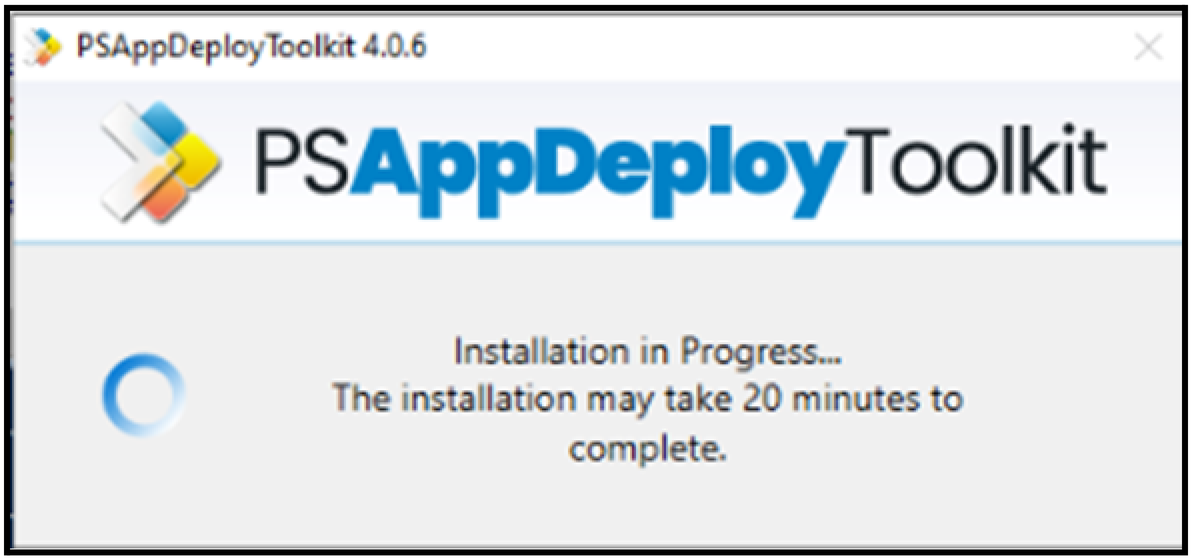
```
ation with message [Installation started].

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>
```

3) **Show-ADTInstallationProgress**
Displays a progress dialog in a separate thread with an updateable custom message. It's a PowerShell function inside Add-AppDevPackage.ps1 which is created to show a graphical progress bar while the app is installing.

Launches a Windows Form with a **progress bar**. It begins the Add-AppxPackage installation of the MSIX bundle. It updates the GUI with output (success/fail). It may also write logs to disk (installation/uninstallation).

Uses: It visually shows progress of MSIX or AppX package installation.
It runs while Add-AppxPackage executes.
It displays success or failure once complete.
It improves the user experience during sideload installs.

Commands: Show-ADTInstallationProgress -StatusMessage 'Installation in Progress...' -WindowLocation 'BottomRight' -NotTopMost



Installation in Progress...
The installation may take 20 minutes to complete.

4) **Show-ADTDialogBox**

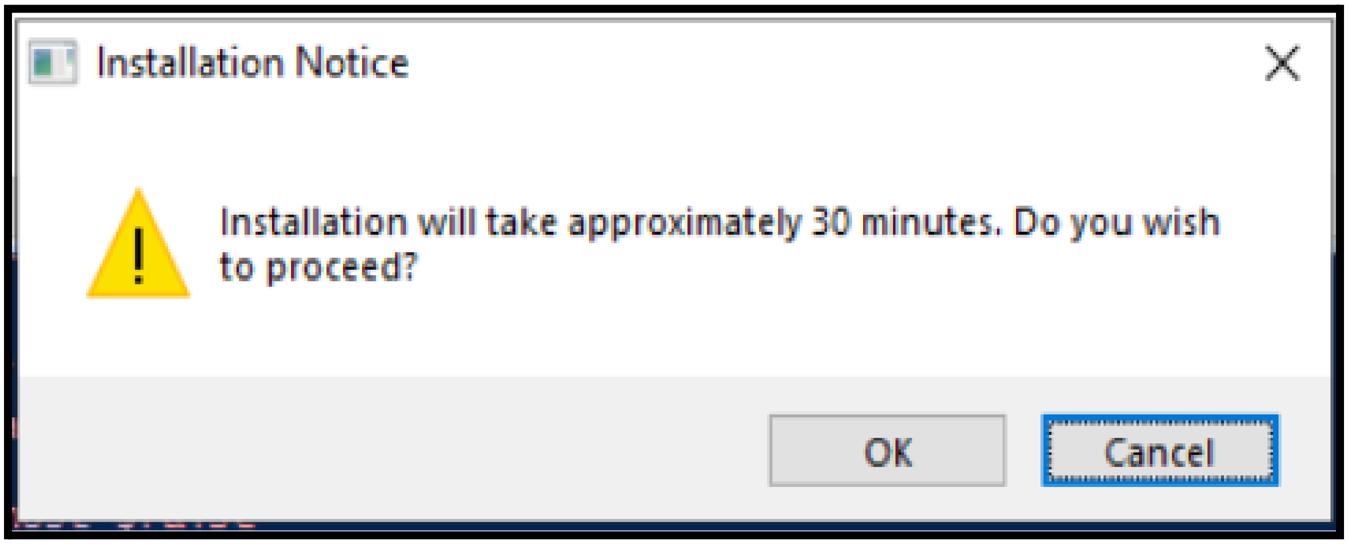
Display a custom dialog box with optional title, buttons, icon, and timeout. The default button is "OK", the default Icon is "None" and the default Timeout is None.

It is a helper function in Add-AppDevPackage.ps1 which is used to show **generic dialog boxes** like alerts, confirmations, errors, etc. with custom buttons, messages and icons.

Uses: It is used to ask the user a Yes/No question.
It is used to show error messages.
It is used to display alerts or warnings.
It returns the user's choice as a result.

If an installation fails or the certificate is missing, the script might call powershell Show-ADTDialogBox "Certificate not installed. Please install the .cer file first."
` "Installation Failed" `

Commands: Show-ADTDialogBox -Title 'Installation Notice' -Text 'Installation will take approximately 30 minutes. Do you wish to proceed?' -Buttons 'OKCancel' -DefaultButton 'Second' -Icon 'Exclamation' -Timeout 600 -Topmost $false


Installation Notice

Installation will take approximately 30 minutes. Do you wish to proceed?

5) Show-ADTBalloonTip

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.

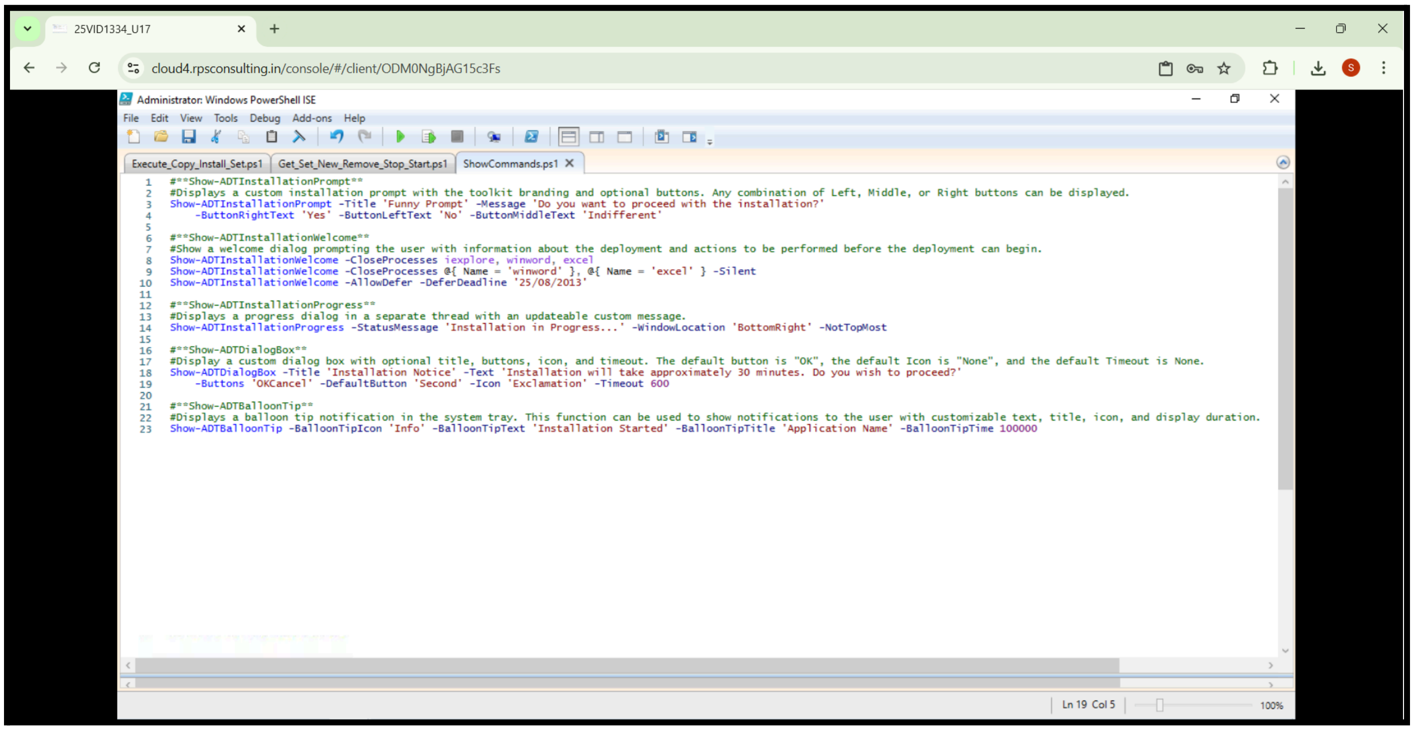
It’s a helper function inside Add-AppDevPackage.ps1 which is used to display a **balloon notification** (system tray toast) when i) the app is installed successfully ii) installation fails iii) a status update is needed. We can think of it as a non-intrusive **system notification**, rather than a pop-up dialog.

Uses: It notifies after installation “App installed successfully”
It shows background information “Installing your app”
It shows error information “Could not install certificate”

It is usually called at the end of Show-ADTInstallationProgress or in error handling logic after Add-AppxPackage fails. It's helpful for scenarios where we want the script to finish quietly. Also, the users may not be looking directly at the console or dialog.

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

Completed


25VID1334_U17
cloud4.rpsconsulting.in/console/#/client/ODM0NgBjAG15c3Fs
Administrator: Windows PowerShell ISE
File Edit View Tools Debug Add-ons Help
Execute_Copy_Install_Set.ps1
Get_Set_New_Remove_Stop_Start.ps1
ShowCommands.ps1
Ln 19 Col 5
100%

```
#**Show-ADTInstallationPrompt**
#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 -Title 'Funny Prompt' -Message 'Do you want to proceed with the installation?'
    -ButtonRightText 'Yes' -ButtonLeftText 'No' -ButtonMiddleText 'Indifferent'

#**Show-ADTInstallationWelcome**
#Show a welcome dialog prompting the user with information about the deployment and actions to be performed before the deployment can begin.
Show-ADTInstallationWelcome -CloseProcesses iexplore, winword, excel
Show-ADTInstallationWelcome -CloseProcesses @{ Name = 'winword' }, @{ Name = 'excel' } -Silent
Show-ADTInstallationWelcome -AllowDefer -DeferDeadline '25/08/2013'

#**Show-ADTInstallationProgress**
#Displays a progress dialog in a separate thread with an updateable custom message.
Show-ADTInstallationProgress -StatusMessage 'Installation in Progress...' -WindowLocation 'BottomRight' -NotTopMost

#**Show-ADTDialogBox**
#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.
Show-ADTDialogBox -Title 'Installation Notice' -Text 'Installation will take approximately 30 minutes. Do you wish to proceed?'
    -Buttons 'OKCancel' -DefaultButton 'Second' -Icon 'Exclamation' -Timeout 600

#**Show-ADTBalloonTip**
#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.
Show-ADTBalloonTip -BalloonTipIcon 'Info' -BalloonTipText 'Installation Started' -BalloonTipTitle 'Application Name' -BalloonTipTime 100000
```

❖ Analysis Result:

The deployment script using PowerShell App Deployment Toolkit (PSADT) executed successfully.

1) **Execution of Script:**
   All prompts, dialogs, progress indicators and system tray notifications worked as expected. The installation completed without errors.
2) **User Interaction:**
   The balloon notifications and the GUI prompts give really good feedback to the end users. It becomes simple for the users to defer installation or proceed immediately based on the options set in the script.
3) **Performance:**
   The process remained stable, with no high CPU / Memory usage or forced reboots.

❖ Technologies/Tools Used:

- **Windows 10/11:** For a safe testing environment.
- **Windows PowerShell ISE:** For writing and testing deployment scripts.
- **PSAppDeployToolkit:** For handling GUI prompts, installation flow and logging.