

Known issues and troubleshooting tips for the MSIX Packaging Tool

1. MSIX Troubleshooter (Advanced Installer)

A popular tool that automatically detects and solves MSIX installation issues.

Key Features:

- Automatic Log Analysis: Scans installation logs to find error causes.
- Smart Event Log Filtering: Highlights only relevant errors.
- Issue Diagnosis Report: Creates a detailed report of the issue.
- Built-in Fixes: Can fix some issues automatically.

2. MSIX Packaging Tool with Package Analyzer (Microsoft)

This built-in tool helps you analyze your MSIX package and suggests improvements.

Key Features:

- Package Analysis: Detects configuration and structure issues.
- Runtime Issue Detection: Suggests how to fix runtime errors.
- Supports GUI and CLI: Offers both Graphical User Interface and Command Line Interface for flexibility.

3. Event Viewer (Manual Log Review)

Built-in Windows tool to manually view MSIX installation logs.

How to use:

- Open Event Viewer.
- Navigate to: Applications and Services Logs > Microsoft > Windows > AppXDeployment-Server
- Review the logs for error codes and descriptions.

4. Other Tools & Techniques

- dumpbin.exe /headers: Command-line tool to check PE headers and identify certificate or file corruption.
- **PowerShell**: Use scripts to access and review AppInstaller interaction logs.
- **MSIXMGR Tool**: Especially useful for Azure Virtual Desktop, helps manage MSIX packages efficiently.

5. Troubleshooting Workflow

1. Identify the Issue

Use MSIX Troubleshooter or Event Viewer to understand the error.

2. Analyze the Logs

Look for specific error codes in logs for targeted diagnosis.

3. Apply Fixes

Use built-in solutions from MSIX Troubleshooter or correct manually using log insights.

4. Re-package and Deploy

Once fixed, repackage your MSIX file and redeploy

Commands

UI Prompts

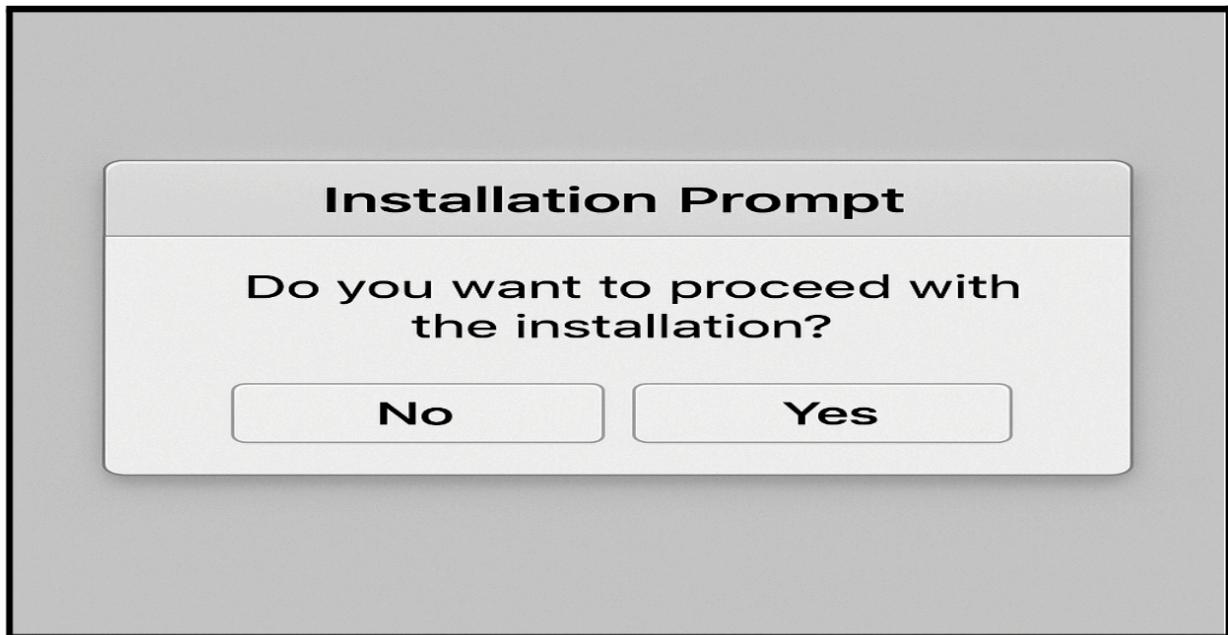
1. Show-InstallationPrompt

It is a browser trigger notification that allows users to install a Progressive web Application to their device.

Show-InstallationPrompt -Message 'Do you want to proceed with the installation?' -ButtonRightText 'Yes' -ButtonLeftText 'No'

`Show-InstallationPrompt -Message 'Do you want to proceed with the installation?' -ButtonRightText 'Yes' -ButtonLeftText 'No'`

Output:



2. Show-InstallationWelcome

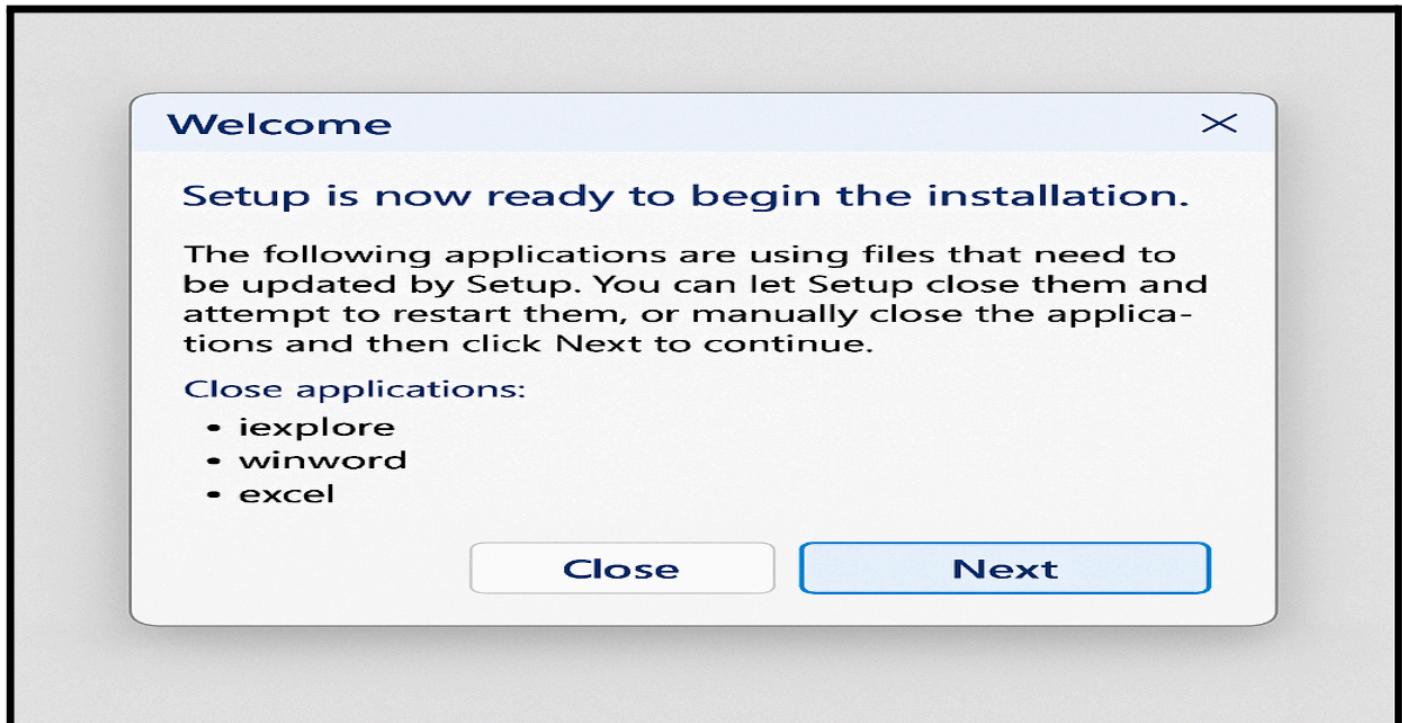
Installation Welcome Prompt is a user interface element displayed after a web app is successfully installed, used to welcome the user and introduce key

features or next steps.

Show-InstallationWelcome -CloseApps 'iexplore,winword,excel'

Show-InstallationWelcome -CloseApps 'winword,excel' -Silent

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

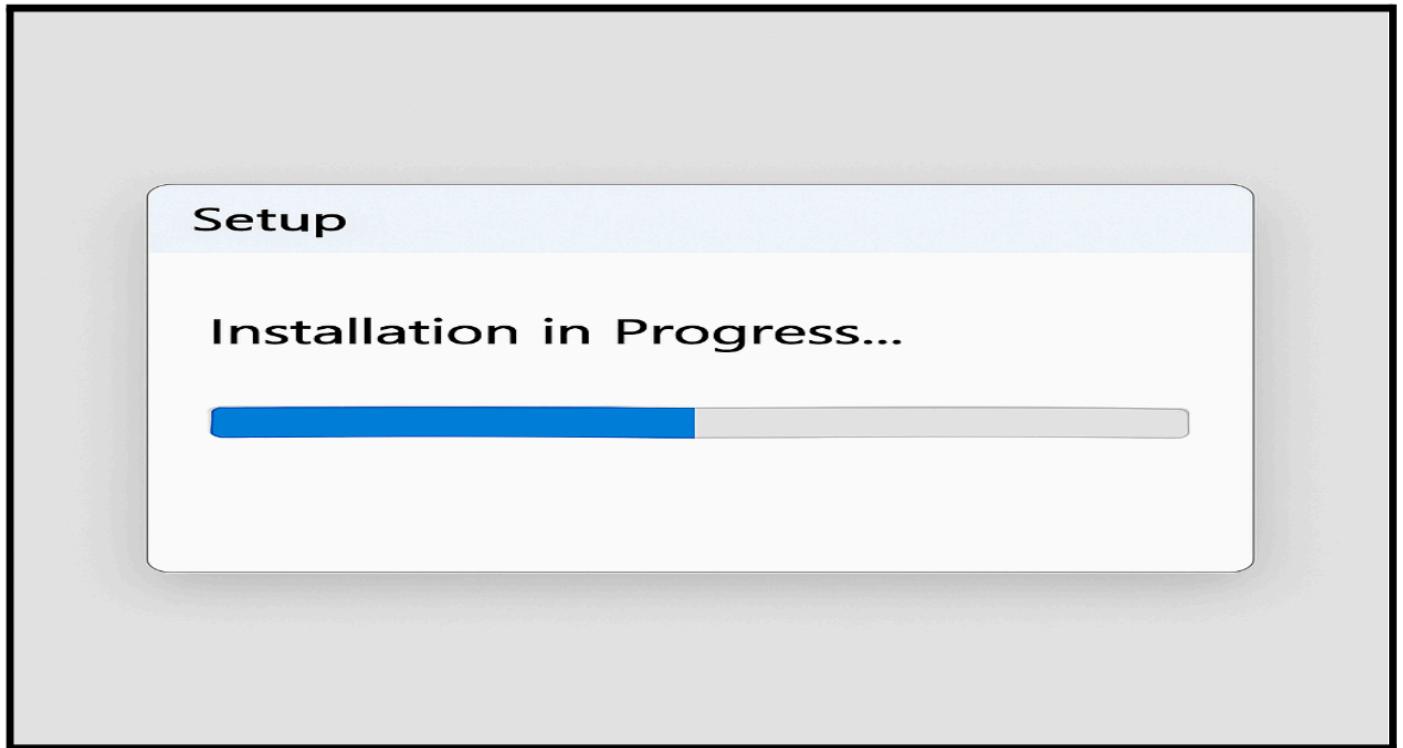


3. Show-InstallationProgress

It is a UI element that visually represents the ongoing status of an application's installation process, helping users understand that installation is in progress and how much is completed.

Show-InstallationProgress -StatusMessage 'Installation in Progress...'

```
Show-InstallationProgress -StatusMessage 'Installation in Progress...'
```



4. Show-DialogBox

Show-Dialog Box is a user interface element that presents information, warnings, or options to users in a separate pop-up window, usually requiring them to respond before continuing.

Show-DialogBox -Title 'Installed Complete' -Text 'Installation has completed. -Icon 'Information'

Please click OK and restart your computer! -Icon 'Information'

```
Show-DialogBox -Title 'Installed Complete' -Text 'Installation has completed. Please click OK and restart your computer.' -Icon 'Information'
```

Install Complete



Installation has completed.
Please click OK and restart
your computer.

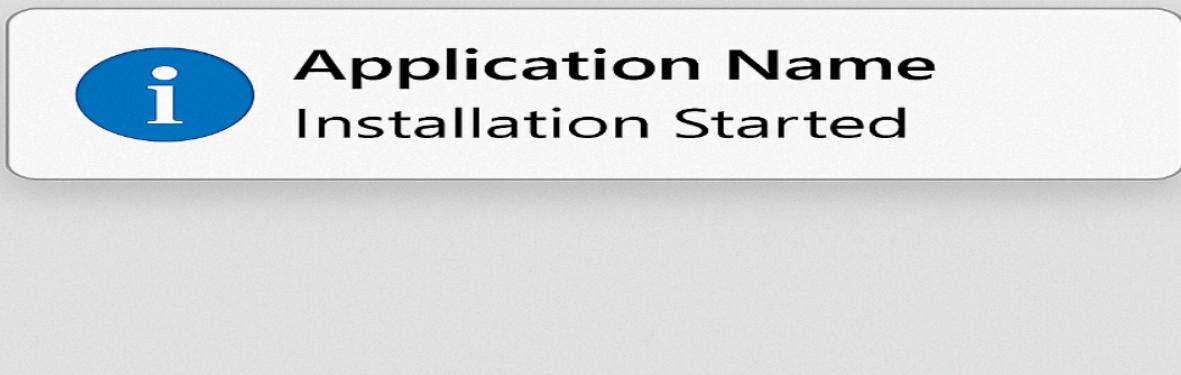
OK

5. Show-Baloontip

Show-Balloon Tip is a small pop-up notification that appears near a system tray icon or UI element to provide brief information, tips, or alerts to the user without interrupting their workflow.

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

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



6. Get-Pending Reboot

PowerShell function or script used to check if a system restart is required after updates, software installations, or system changes.

Get-PendingReboot

(Get-PendingReboot).IsSystemRebootPending

Get-PendingReboot
(Get-PendingReboot).IsSystemRebootPending

```
(Get-PendingReboot).IsSystemRebootPending  
True
```