TunedIn:Leeds

PSAppDeployToolkit 101

TUNEDIN
MEETUPS

adaptiva

Introduction - Dan Gough

- Software Engineer @ Patch My PC
- PSAppDeployToolkit developer
- Former MVP for App-V
- Blog @ packageology.com
- x.com/packageologist
- linkedin.com/in/danielgough

TUNEDIN MEETUPS

TunedIn: Leeds

adaptiva

What is PSAppDeployToolkit?

- An open-source PowerShell framework for creating application deployment scripts
- A set of easy-to-use functions to simplify deployment tasks
- A standardised way to install, uninstall and repair applications

Features

- Standardised logging of all functions and MSI operations
- Optionally exit the script if specific installation steps fail
- Branded GUI to manage prompts, deferrals, and reboots
- Ensure applications are closed before updating and prevent them from being launched until complete
- Automatic detection of PowerPoint presentation mode
- Uninstall applications based on name, publisher, version, etc

Features

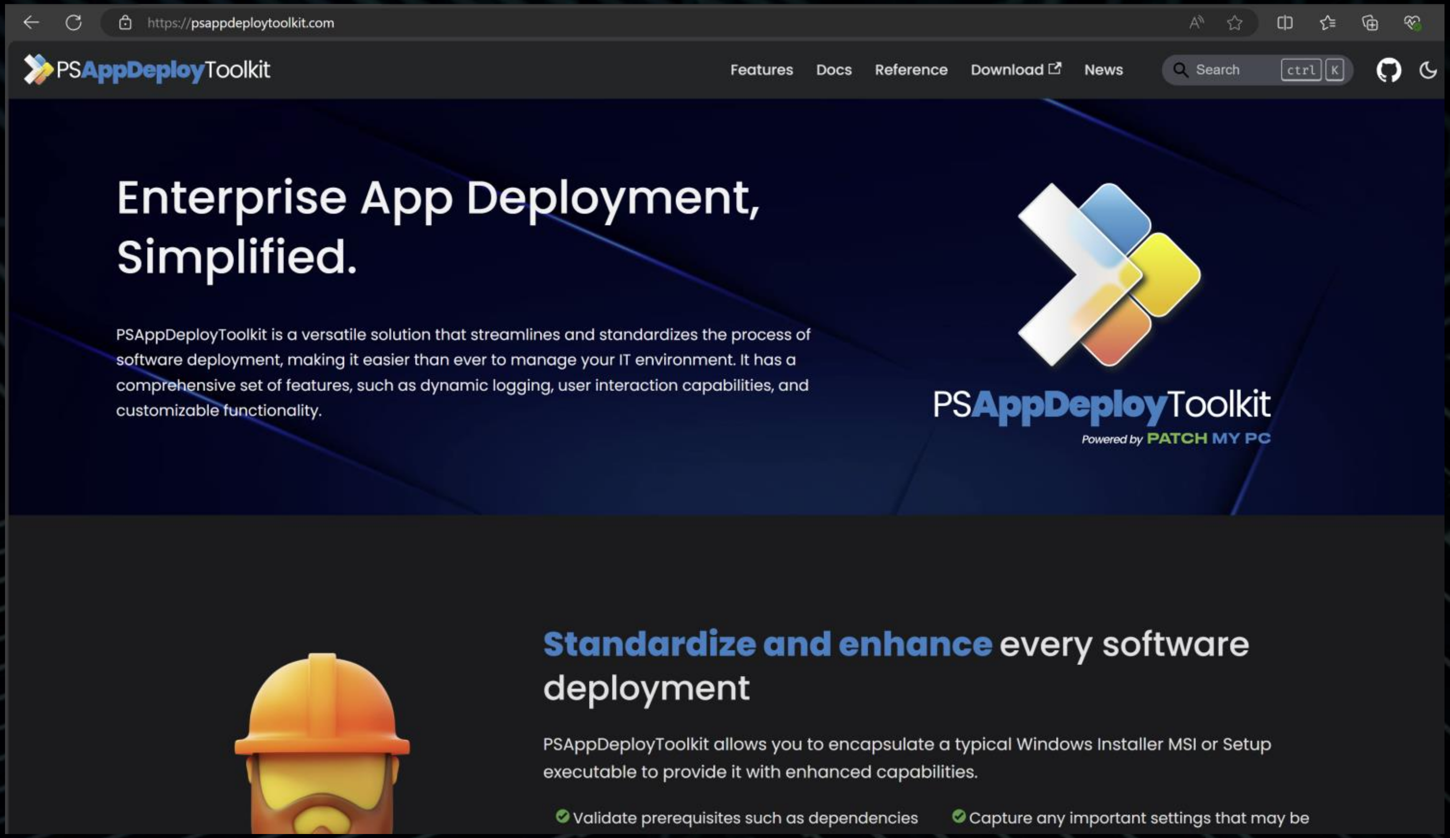
- Manage user files and registry from the System account
- Run processes as the current user from the System account
- Manage Active Setup and auto-run for the current user
- Pin apps to the Start Menu or Task Bar
- Create shortcuts
- Manage extensions for Microsoft Edge
- Cache installers locally before execution

Benefits

- Reduces the learning curve with PowerShell scripting
- A robust and battle-tested deployment framework
- A widely adopted standard for enterprise deployment
- Simplifies deployment troubleshooting
- Customizable user interface – both images and text
- Extensive community support – GitHub, Forum, Reddit, Discord

Getting Started

https://psappdeploytoolkit.com



Enterprise App Deployment, Simplified.

PSAppDeployToolkit is a versatile solution that streamlines and standardizes the process of software deployment, making it easier than ever to manage your IT environment. It has a comprehensive set of features, such as dynamic logging, user interaction capabilities, and customizable functionality.

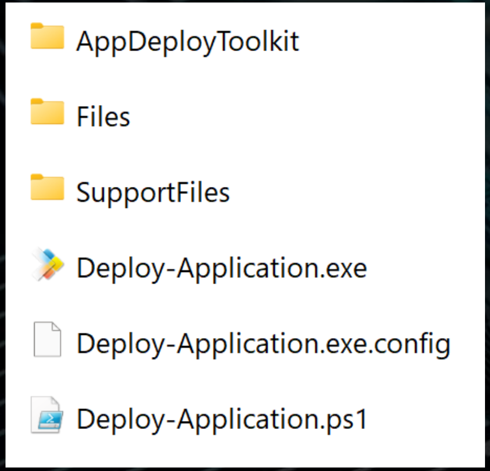
Standardize and enhance every software deployment

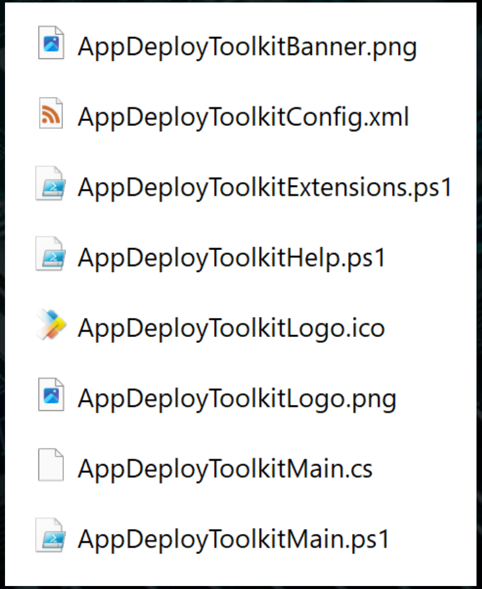
PSAppDeployToolkit allows you to encapsulate a typical Windows Installer MSI or Setup executable to provide it with enhanced capabilities.

✓ Validate prerequisites such as dependencies ✓ Capture any important settings that may be

Getting Started

Download the latest release from:
https://github.com/PSAppDeployToolkit/PSAppDeployToolkit/releases


AppDeployToolkit
Files
SupportFiles
Deploy-Application.exe
Deploy-Application.exe.config
Deploy-Application.ps1

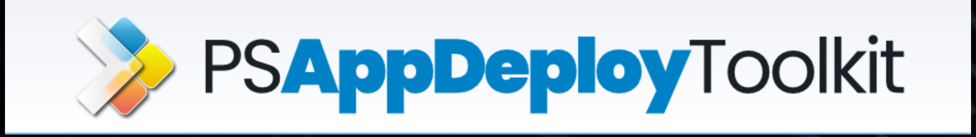

AppDeployToolkitBanner.png
AppDeployToolkitConfig.xml
AppDeployToolkitExtensions.ps1
AppDeployToolkitHelp.ps1
AppDeployToolkitLogo.ico
AppDeployToolkitLogo.png
AppDeployToolkitMain.cs
AppDeployToolkitMain.ps1

AppDeployToolkitConfig.xml

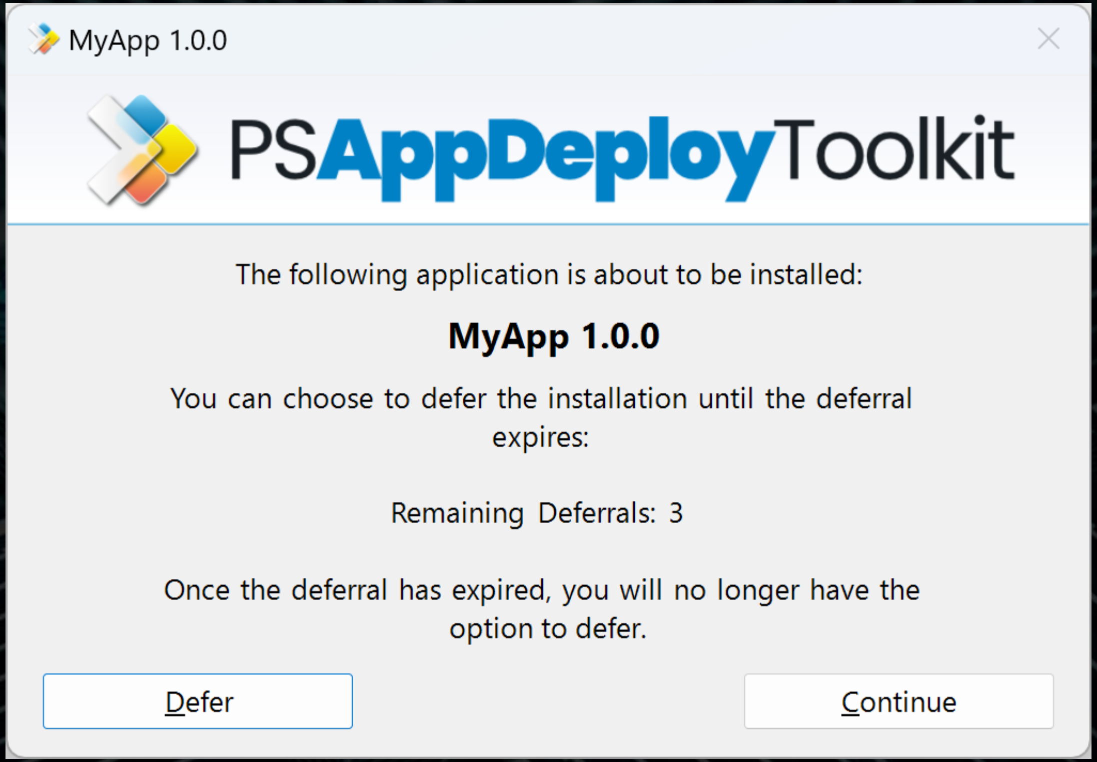
- <Toolkit_RequireAdmin>True</Toolkit_RequireAdmin>
- <Toolkit_LogPath>$envWinDir\Logs\Software</Toolkit_LogPath>
- <Toolkit_LogAppend>True</Toolkit_LogAppend>
- <Toolkit_UseRobocopy>True</Toolkit_UseRobocopy>
- <MSI_InstallParams>REBOOT=ReallySuppress /QB-! </MSI_InstallParams>
- <ShowBalloonNotifications>True</ShowBalloonNotifications>

Corporate Branding

AppDeployToolkitBanner.png:



AppDeployToolkitLogo.png / AppDeployToolkitLogo.ico:



The following application is about to be installed:

MyApp 1.0.0

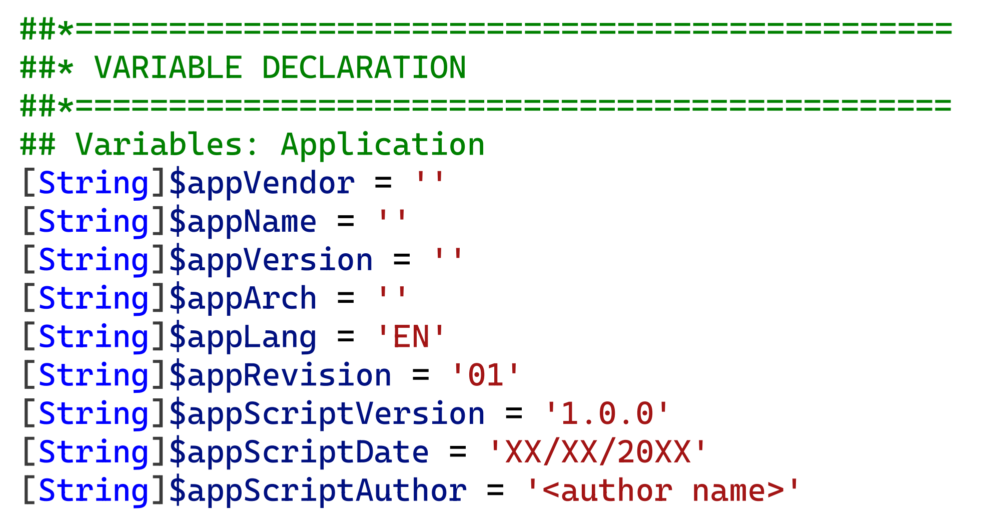
You can choose to defer the installation until the deferral expires:

Remaining Deferrals: 3

Once the deferral has expired, you will no longer have the option to defer.

Deploy-Application.ps1

- Set your variables for vendor, name, version, etc:

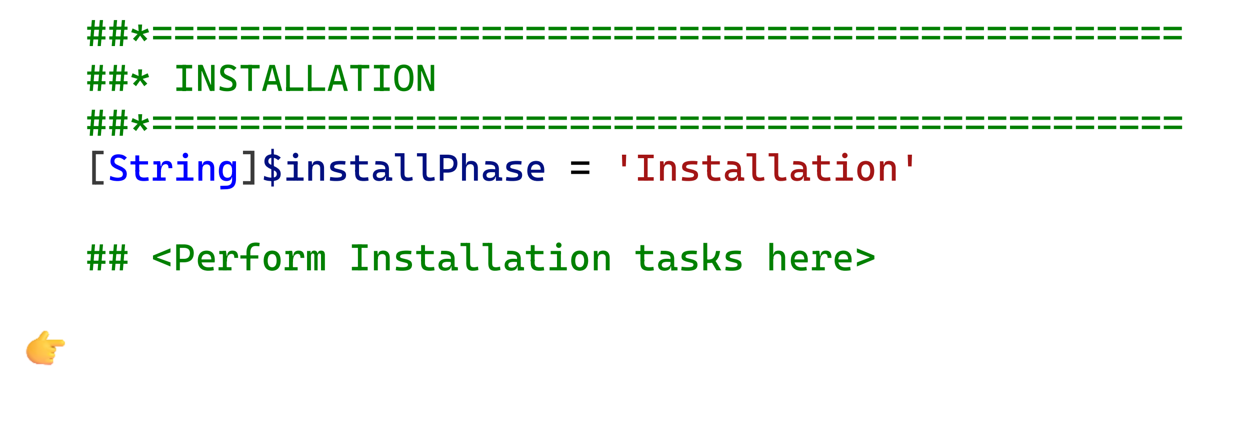


```
##*===============================================
##* VARIABLE DECLARATION
##*===============================================
## Variables: Application
[String]$appVendor = ''
[String]$appName = ''
[String]$appVersion = ''
[String]$appArch = ''
[String]$appLang = 'EN'
[String]$appRevision = '01'
[String]$appScriptVersion = '1.0.0'
[String]$appScriptDate = 'XX/XX/20XX'
[String]$appScriptAuthor = '<author name>'
```

- Or leave $appName empty to activate Zero-Config mode

Deploy-Application.ps1

- There are multiple sections where you can insert your code:
  - Pre-Installation
  - Installation
  - Post-Installation
  - Pre-Uninstallation
  - Uninstallation
  - Post-Uninstallation



```
##*===============================================
##* INSTALLATION
##*===============================================
[String]$installPhase = 'Installation'

## <Perform Installation tasks here>
```

TunedIn: Leeds

Toolkit Variables

- Run Get-Variable (or alias gv) to get all currently defined variables after running AppDeployToolkitMain.ps1
- Full reference in the docs: https://psappdeploytoolkit.com/docs/variables
- Variables are created for many useful folders:
  - $dirFiles
  - $dirSupportFiles
  - $envCommonDesktop
  - $envCommonStartMenuPrograms

Pre-Installation Tasks

- Perform any pre-flight deployment checks
  - Is a higher version already installed?
  - Are all dependencies installed?
  - Is the application compatible with the OS?
- Display user interface
  - Optional – UI functions are skipped if toolkit is run in silent mode
  - Display a welcome dialog giving the user the chance to defer
- Uninstall existing applications / specific versions
  - For apps that do not upgrade properly, or to downgrade
  - Remove per-user installations when installing a per-machine installation
  - Remove setup.exe install when installing an MSI
  - Remove apps based on name matching

Pre-Installation Functions

Get-InstalledApplication

```
Get-InstalledApplication -Name 'Zoom*' -WildCard

Get-InstalledApplication -ProductCode '{334503B4-0A36-45A2-8206-A6B37A1F8B5B}’

Get-InstalledApplication -Name 'Zoom*' -WildCard | Where-Object { $_.Publisher -eq 'Zoom' -and
$_.Is64BitApplication -eq $true -and [version]$_.DisplayVersion -lt [version]$appVersion }
```

Remove-MSIApplications

```
Remove-MSIApplications -Name 'Java’

Remove-MSIApplications -Name 'Java' -FilterApplication @(, @('Publisher', 'Oracle Corporation', 'Exact'))
```

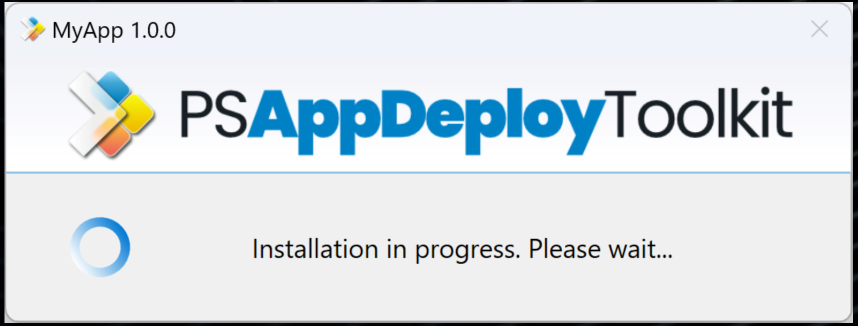
Pre-Installation Functions

Show-InstallationWelcome

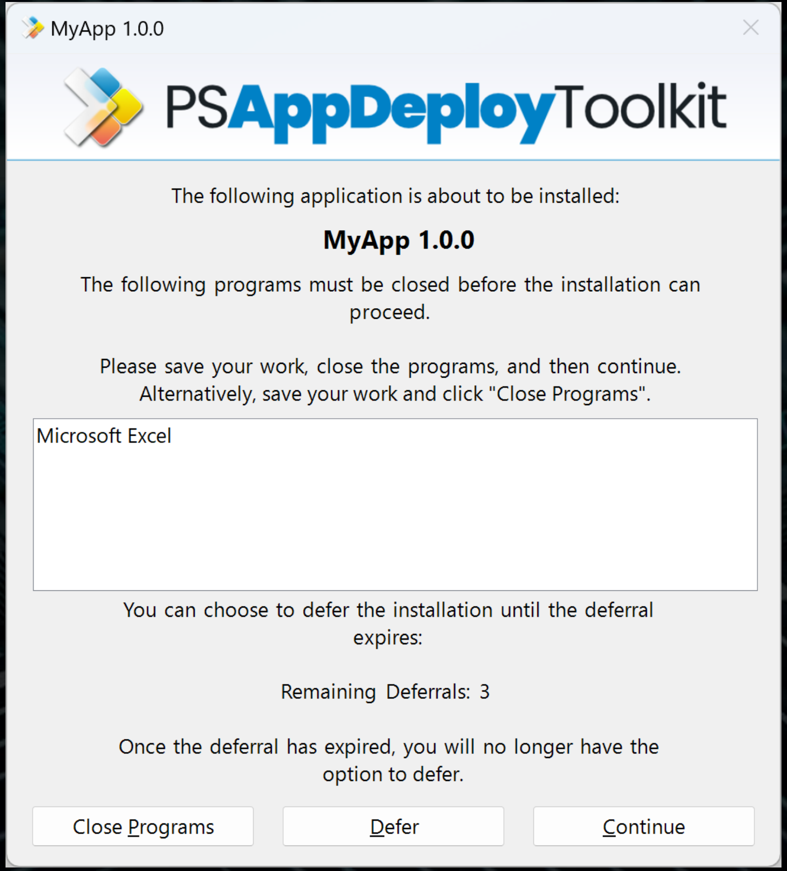
```
Show-InstallationWelcome -CloseApps 'excel'
-AllowDeferCloseApps -DeferTimes 3
-PersistPrompt -MinimizeWindows $false
```

Show-InstallationProgress

```
Show-InstallationProgress -TopMost $false
```



Installation in progress. Please wait...



The following application is about to be installed:

MyApp 1.0.0

The following programs must be closed before the installation can proceed.

Please save your work, close the programs, and then continue. Alternatively, save your work and click "Close Programs".

Microsoft Excel

You can choose to defer the installation until the deferral expires:

Remaining Deferrals: 3

Once the deferral has expired, you will no longer have the option to defer.

TUNEDIN MEETUPS

TunedIn: Leeds

adaptiva

Installation Functions

Execute-Process

```
Execute-Process -Path 'vcredist_x64.exe' -Parameters '/quiet /norestart' -WaitForMsiExec -ExitOnProcessFailure $false
```

Execute-ProcessAsUser

```
Execute-ProcessAsUser -Path "$runasUserProfile\AppData\Local\PerUserApp\uninstall.exe" -Parameters '/q'
-ContinueOnError $true -Wait
```

Execute-MSI

```
Execute-MSI -Action 'Install' -Path 'MyApp.msi' -Transform 'MyApp.mst' -AddParameters "LICENSEKEY=$LicenseKey"
```

Post-Installation Tasks

- Create / Delete shortcuts
- Copy files (to a fixed location or to all user profiles)
- Set registry keys (to HKLM or to HKCU for all users)
- Set permissions on files or registry keys
- Create an active setup (including running it immediately for the logged-in user)

Post-Installation Functions

Copy-File

```
Copy-File -Path "$dirFiles\config.xml" -Destination "$envProgramFile\MyApp"
```

Remove-File

```
Remove-File -Path "$envCommonDesktop\MyApp.lnk","$envCommonStartMenuPrograms\MyApp\Uninstall.lnk"
```

Copy-FileToUserProfiles

```
Copy-FileToUserProfiles -Path "$dirFiles\config.xml" -Destination 'AppData\Roaming\MyApp'
```

Post-Installation Functions

New-Shortcut

```
New-Shortcut -Path "$envCommonStartMenuPrograms\MyApp\MyApp.lnk" -TargetPath "$envProgramFile\MyApp\MyApp.exe"
```

Set-RegistryKey

```
Set-RegistryKey -Key 'HKLM\Software\MyApp' -Name 'AutoUpdate' -Value 0 -Type Dword
```

Invoke-HKCURegistrySettingsForAllUsers

```
[scriptblock]$HKCURegistrySettings = {
    Set-RegistryKey -Key 'HKCU\Software\MyApp' -Name 'AutoUpdate' -Value 0 -Type DWord -SID $UserProfile.SID
}
Invoke-HKCURegistrySettingsForAllUsers -RegistrySettings $HKCURegistrySettings
```

Post-Installation Functions

Set-ItemPermission

```
Set-ItemPermission -Path "$envProgramFiles\MyApp" -User '*S-1-5-32-545' -Permission Modify -Inheritance
ObjectInherit, ContainerInherit
```

Set-ActiveSetup

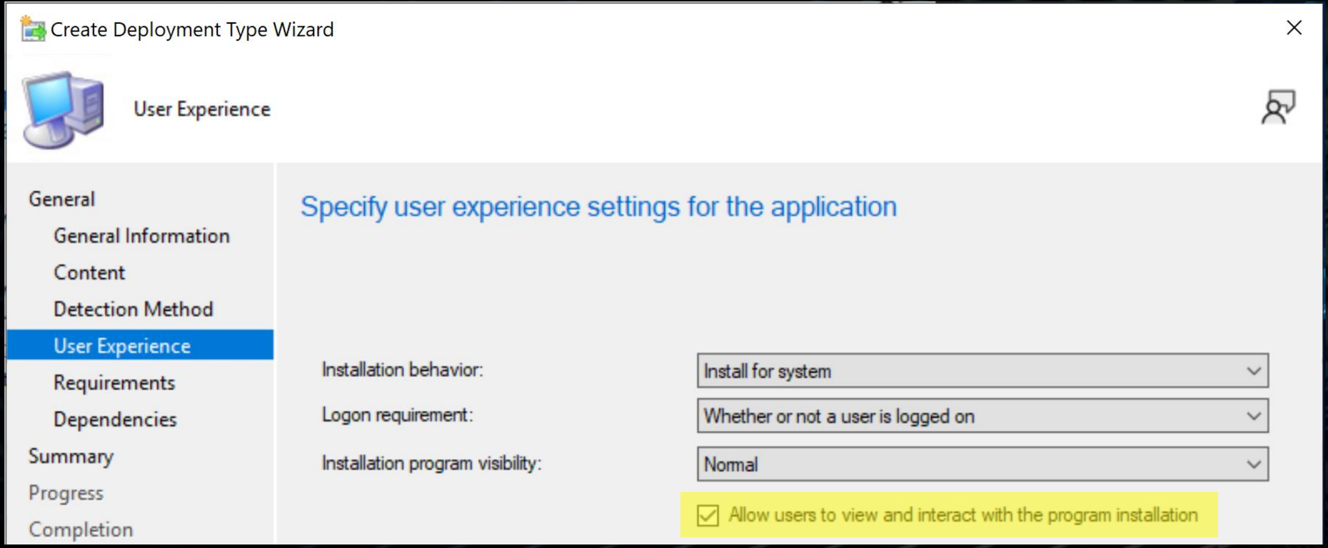
```
Set-ActiveSetup -Key 'MyApp' -Description 'My Active Setup' -Version '1,0' -StubExePath
"$envProgramFiles\MyApp\MyApp.exe" -Arguments '/config'
```

Deploying Your Script

- To perform a basic interactive install, simply run Deploy-Application.ps1
- Deploy-Application.exe provides a convenient way to launch the toolkit without displaying the console to the user
- Additional switches are available:
  - Deploy-Application.exe -DeploymentType Install -DeployMode Silent
  - Deploy-Application.exe -DeploymentType Uninstall -DeployMode Silent
  - Deploy-Application.exe -DeploymentType Repair -DeployMode Interactive
- To pass 3010 exit codes back to ConfigMgr / Intune, add -AllowRebootPassThru

Deploying Your Script

- If using ConfigMgr and you want to enable the UI, select the option below:



Create Deployment Type Wizard

User Experience

Specify user experience settings for the application

Installation behavior: Install for system

Logon requirement: Whether or not a user is logged on

Installation program visibility: Normal

☑ Allow users to view and interact with the program installation

- Make sure that your installation does not provide any access to a file or web browser, since it will be running as System

TUNEDIN MEETUPS

TunedIn: Leeds

adaptiva

Deploying Your Script

- Intune does not have this same option available, and runs all System context installers hidden
- ServiceUI.exe can be used to make the toolkit visible to the user when required:

```
ServiceUI.exe -process:explorer.exe Deploy-Application.exe
-DeploymentType Install -DeployMode Interactive
```

- Invoke-ServiceUI.ps1 is provided to further enhance this

PSAppDeployToolkit

Demo Time!

TUNEDIN MEETUPS

TunedIn: Leeds

adaptiva

Coming Soon - v4.0!

- Rewritten as a PowerShell Module
- New function names and parameters
- Compatibility mode to allow re-use of existing v3.x deployment scripts
- Modern Windows 11 / Fluent style UI with dark mode support!
- Minimum supported PowerShell versions: 5.1 / 7.4
- Minimum supported OS: Windows 10 22H2 / Server 2016
- Preview release expected Q3 2024

Q&A

Thank you to our Sponsor

adaptiva™