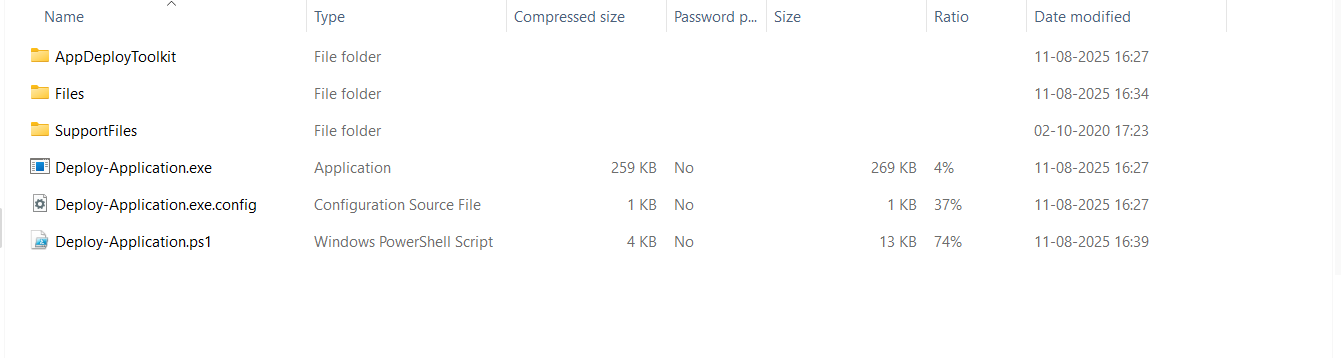
# PowerShell App Deployment Toolkit (PSADT)

## 1. Basic Platform Value, Concepts, and Usage

The PowerShell App Deployment Toolkit (PSADT) is a framework that makes application deployment easier, especially in enterprise environments. It uses PowerShell scripts to handle software installation, updates, and uninstallation in a consistent way. The toolkit provides predefined functions that handle common tasks like logging, showing user prompts, closing running apps, and running installers silently.

## 2. Overview of the Tool Folder Structure

When you download PSADT, you'll see a folder structure like this:  
**- AppDeployToolkit:** Contains the main script engine and function definitions.  
**- Files:** Place your application installers and related files here.  
**- SupportFiles:** Store any helper files or resources.  
**- Deploy-Application.ps1:** The main script you edit for deployment logic.  
**- Deploy-Application.exe / .cmd:** Launchers for running the script.



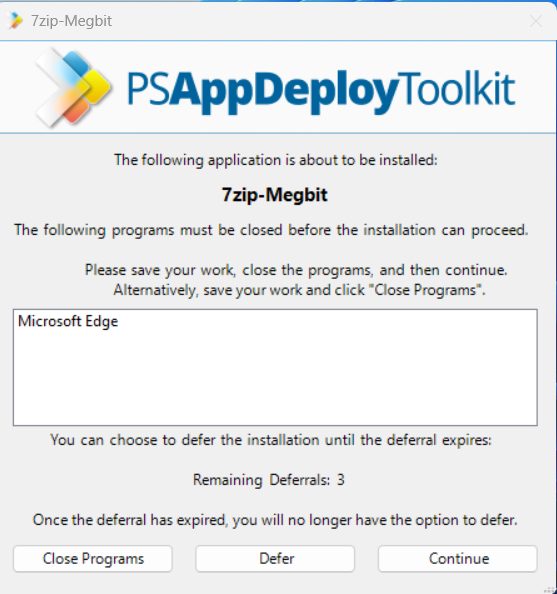
## 3. Tool Configuration

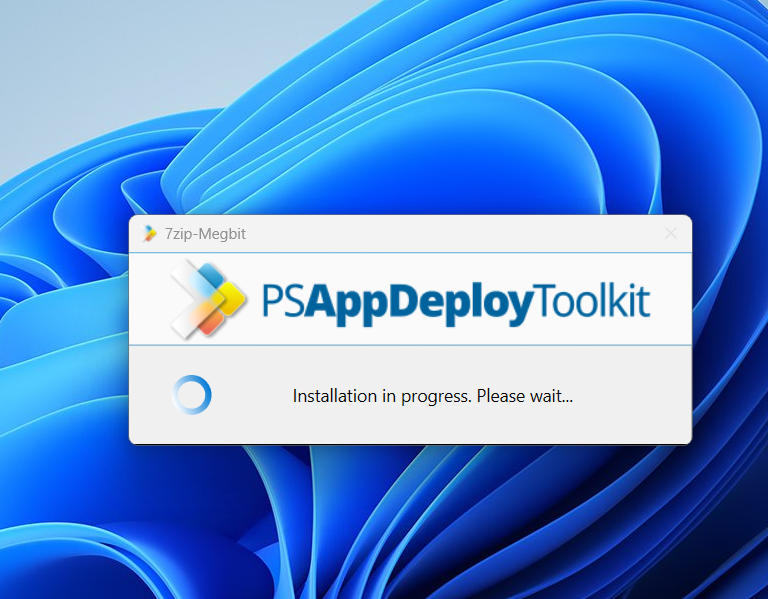
You configure PSADT by editing Deploy-Application.ps1. You specify what happens during installation and uninstallation. You can also adjust AppDeployToolkitConfig.xml to change logging paths, UI behavior, and other defaults.

## 4. Template Script

The Deploy-Application.ps1 file comes preloaded with sections for:  
**- Pre-Installation:** Close apps, show messages, prepare system.  
**- Installation:** Run your MSI, EXE, or MSP.

**Installation commandline :** PS C:\Users\adity> C:\Users\adity\Downloads\Toolkit\Deploy-Application.exe Install

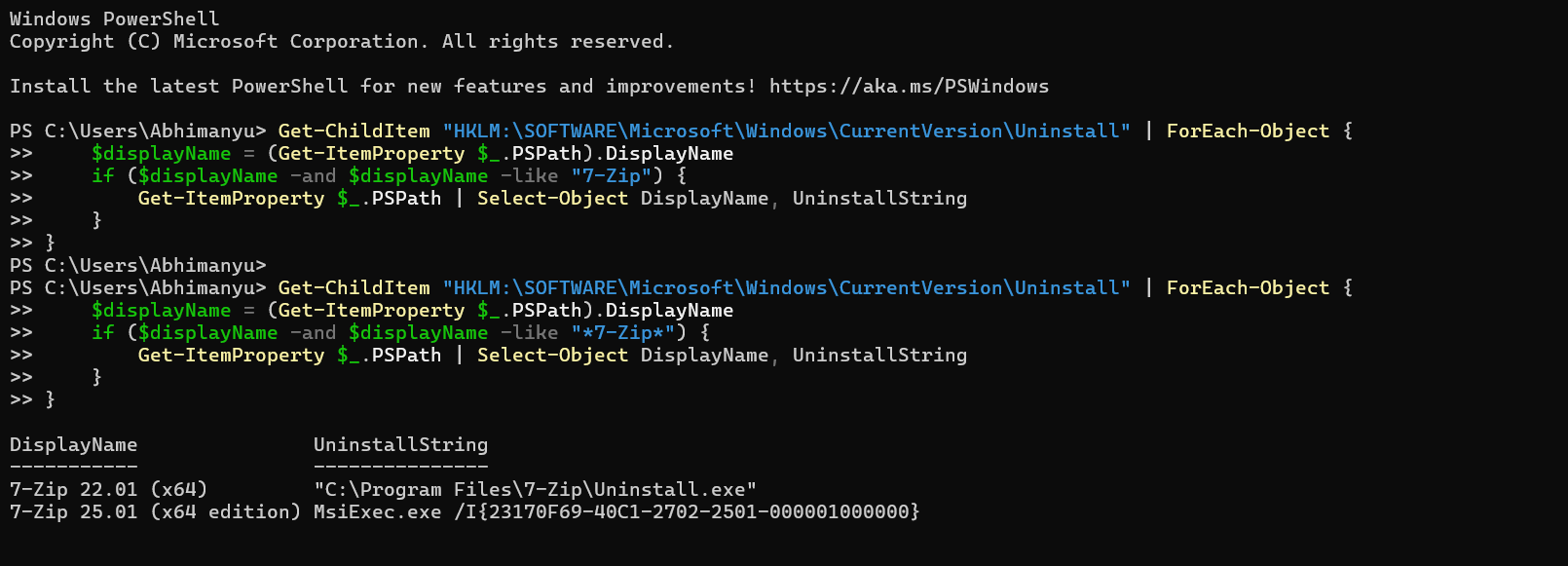


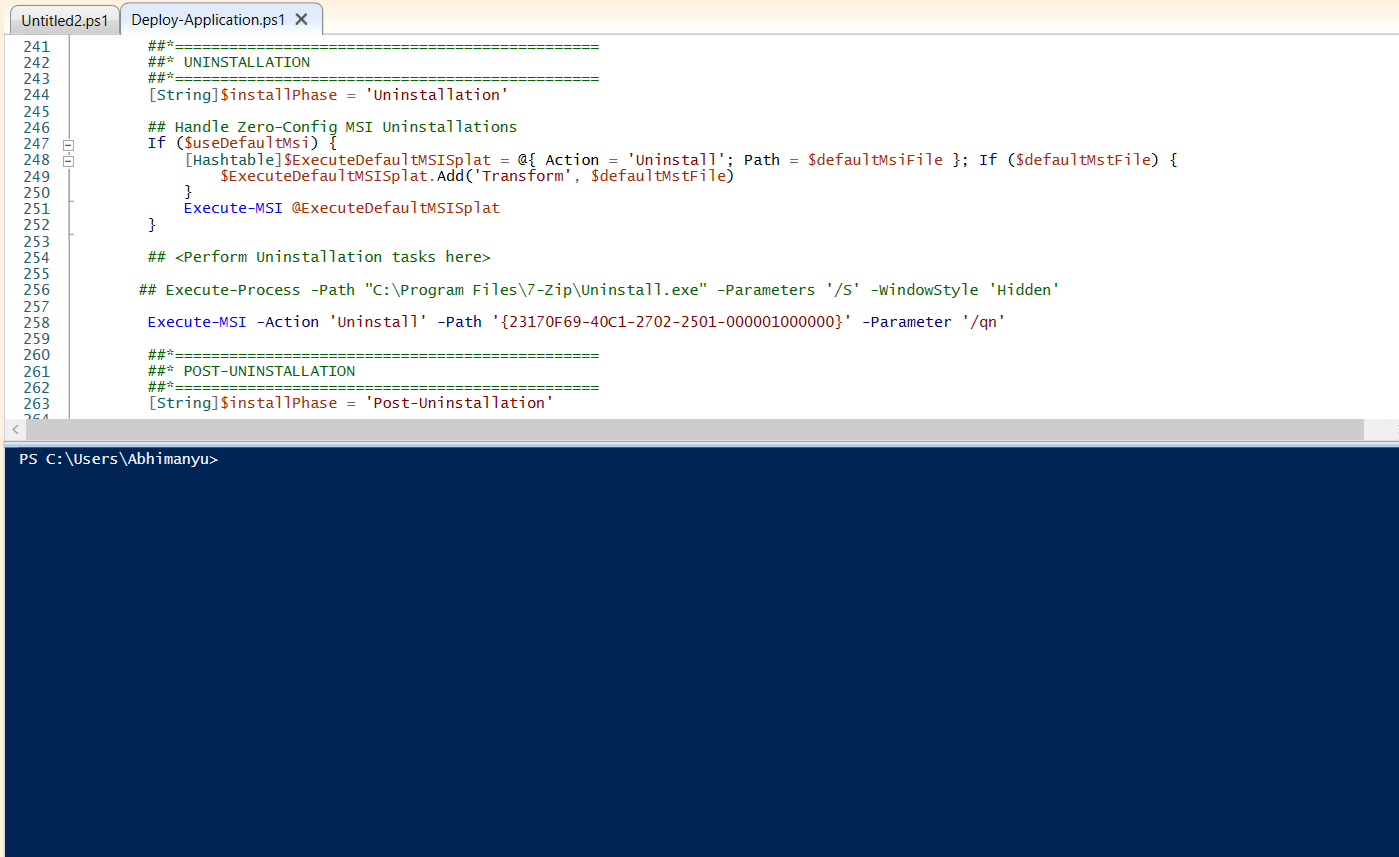


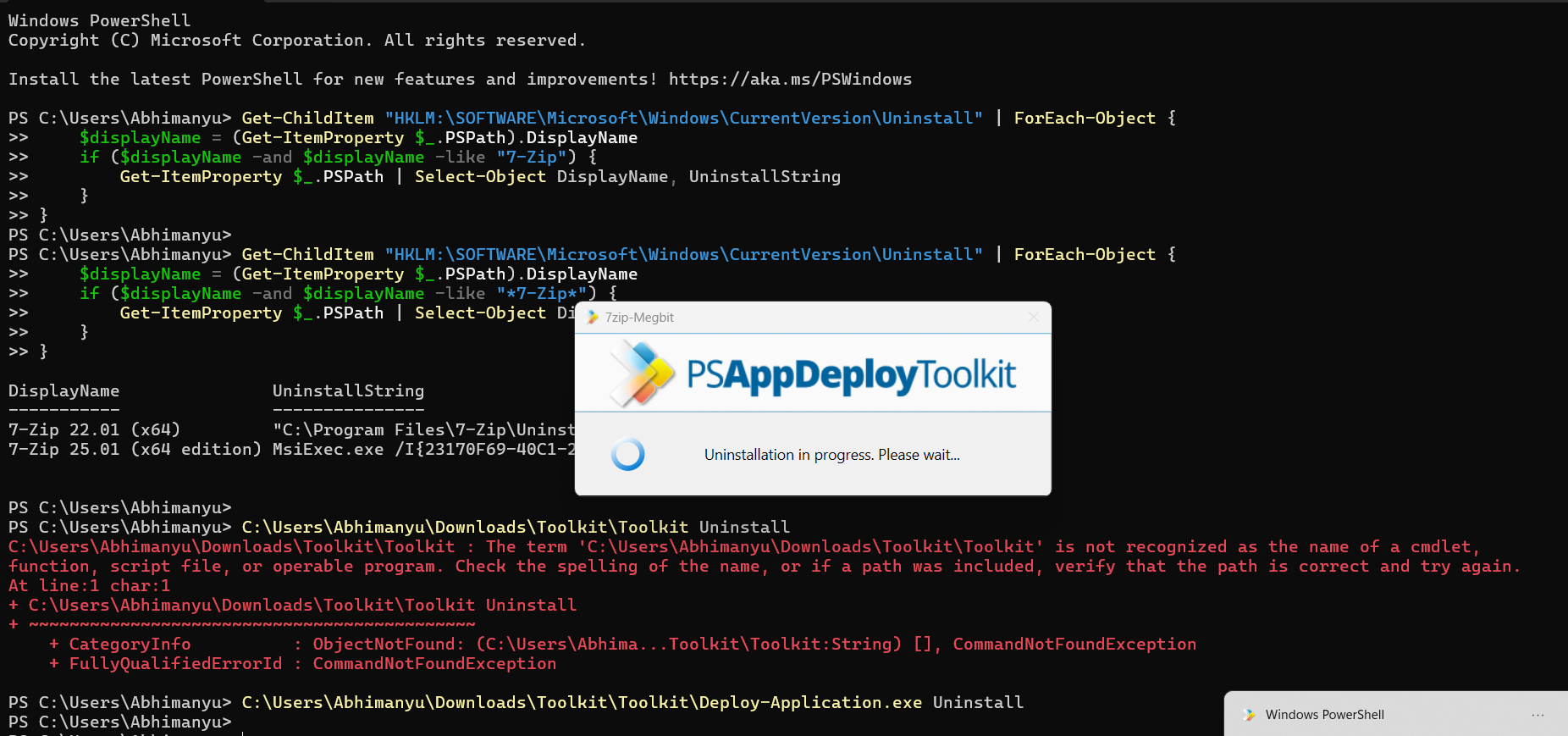
**- Post-Installation:** Show completion message, cleanup.

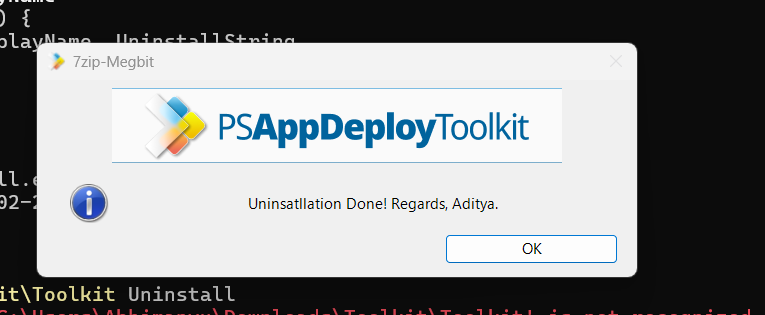
**- Uninstallation:** Similar steps for removing software.

**Uninstallation Commandline :** PS C:\Users\adity> C:\Users\adity\Downloads\Toolkit\Deploy-Application.exe Uninstall









## 5. Predefined Environment Variables and Usage

PSADT defines several environment variables you can use directly in your script:  
**- $appName:** Name of your application.  
**- $installPhase:** Indicates the phase (Pre-Install, Install, Post-Install).  
**- $dirFiles:** Path to the Files folder.  
**- $dirSupportFiles:** Path to the SupportFiles folder.  
These variables save you from hardcoding paths.

## 6. Commands for Installation and Uninstallation

The toolkit includes commands that simplify installs and uninstalls:  
**- Execute-MSI:** Runs an MSI installer or uninstaller with logging.  
**- Execute-Process:** Runs any EXE installer or script.  
**- Execute-MSP:** Runs an MSP patch file.  
**- Install-MSUpdates:** Installs Microsoft Updates.

## 7. Execute-MSI - Installation and Uninstallation

**To install an MSI:**  
Execute-MSI -Action Install -Path 'MyApp.msi'  
**To uninstall an MSI:**  
Execute-MSI -Action Uninstall -Path '{MSI-PRODUCT-CODE}'  
PSADT automatically handles silent parameters and logging.

## 8. Execute-Process

**Use this when installing EXE files:**  
Execute-Process -Path 'setup.exe' -Parameters '/silent /norestart'  
It works for scripts, batch files, and other executable tools as well.

## 9. Execute-MSP

**To apply a Microsoft patch:**  
Execute-MSP -Path 'update.msp'  
This will run the patch with proper silent parameters and logging.

## 10. Install-MSUpdates

**This command installs Microsoft updates (.msu or .cab files):**  
Install-MSUpdates -MSUFile 'Windows10-KB123456.msu'  
It's useful for pushing OS updates during deployments.