Name-Manojit Kumar Das

User id-34873

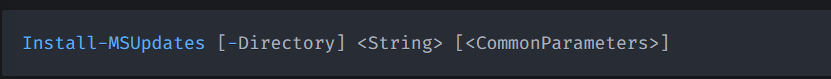
Mail id:-dasmonojit026@gmail.com

**ASSIGNMENTS-10**

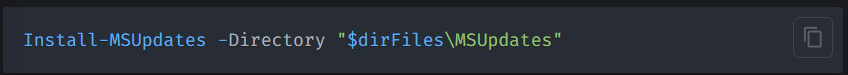
**:Install-MSUpdates:**

In PSAppDeployToolkit, Install-MSUpdates is a helper function used to install Microsoft Update (.msp) patch files.

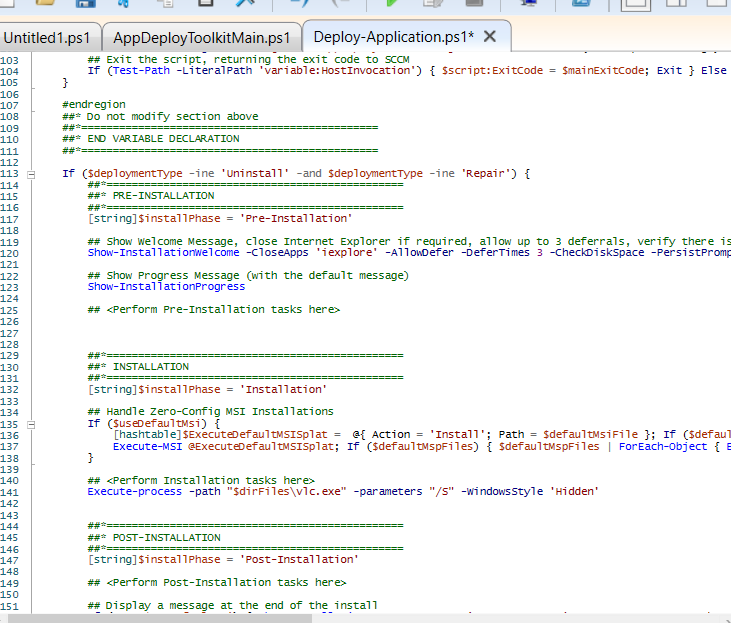
* MSP file = *Microsoft Patch*, usually used to update, fix, or upgrade software that was originally installed via an MSI (Windows Installer package).
* These updates are not for Windows itself (like Windows Update), but for MSI-installed applications (e.g., Adobe Reader, Office, etc.).

**SYNTAX:-**

**EXAMPLE:-**



**:Execute-MSI - Installation and Uninstallation:**

In PSAppDeployToolkit, Execute-MSI is a function that makes installing, uninstalling, and repairing MSI-based applications easier by wrapping the msiexec command with standard logging, error handling, and options.

**Installation:-**

**Uninstallation:-**

Using the MSI file:

Execute-MSI -Action Uninstall -Path "$dirFiles\VLC.msi"

Using the Product GUID:

Execute-MSI -Action Uninstall -Path "{12345678-ABCD-1234-ABCD-1234567890AB}"

**:EXECUTE PROCESS:**

In **plain PowerShell** (without PSAppDeployToolkit), there’s no native cmdlet called Execute-Process.  
It’s actually a **custom function** provided by the **PowerShell App Deployment Toolkit (PSAppDeployToolkit)** — not part of the built-in PowerShell command set.

**Core Features**

1. **Path Handling**  
   Can run executables located in your script’s $dirFiles folder or any full path.
2. **Arguments Support**  
   Lets you pass silent switches easily, e.g., /S, /quiet, /norestart.
3. **Window Control**  
   Can show, hide, minimize, or maximize the process window.
4. **Wait Handling**  
   Automatically waits for the process to complete before moving on (unless told not to).
5. **Exit Code Handling**  
   Logs the return code, compares against known “success” codes, and handles failures gracefully.
6. **Logging**  
   Every call is logged into PSAppDeployToolkit’s log files for auditing.

**:ACTIVE SETUP VERSION:**

In Windows, Active Setup is a mechanism used to run commands for each user the first time they log on, typically for installing or configuring per-user settings of an application.

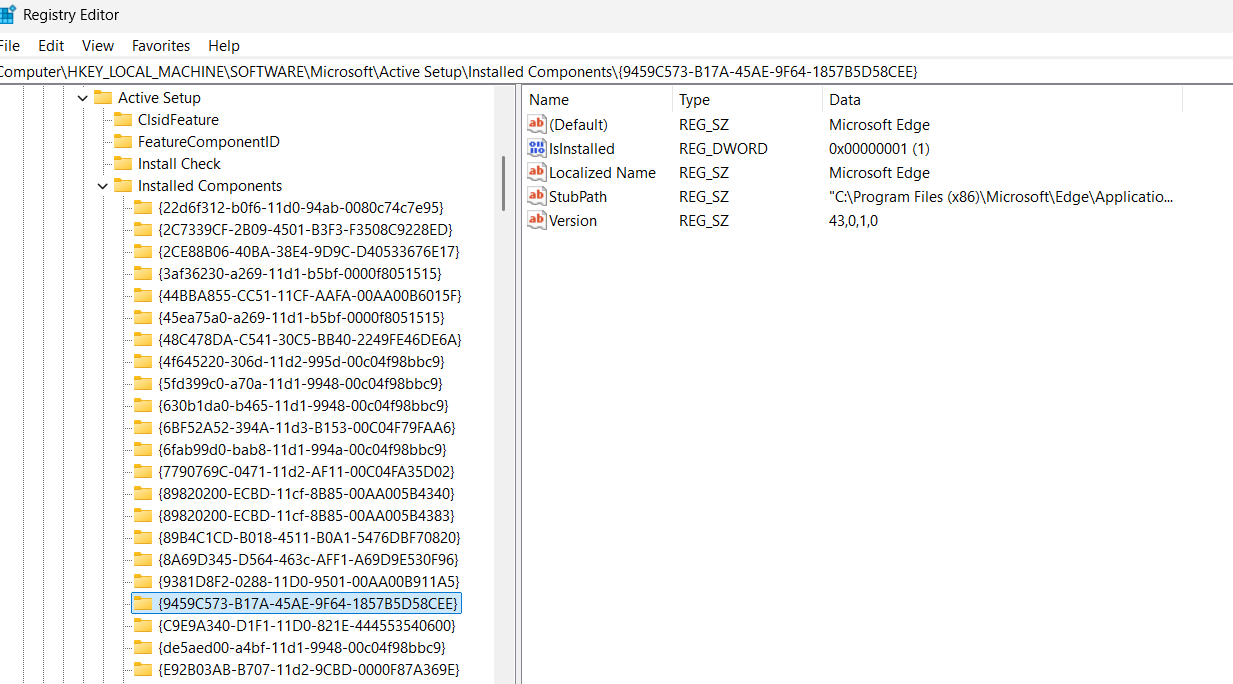
The Active Setup version in PowerShell usually refers to the Version value stored in the registry under the Active Setup component’s key.  
It’s used to determine whether the Active Setup command should run again for a user.

**How It Works:-**

HKLM\SOFTWARE\Microsoft\Active Setup\Installed Components\<GUID or AppName>

HKLM\SOFTWARE\WOW6432Node\Microsoft\Active Setup\Installed Components\<GUID or AppName>

**Example:-**

****

**:Copy/Delete/Move File and Folders in psadt:**

**1. Copying Files and Folders:**

* Copy-ADTFile: This function is designed to copy files during a PSADT deployment. It offers features like error handling and logging specific to the toolkit.

|  |
| --- |
| Copy-ADTFile -Path "$dirSupportFiles\YourFile.txt" -Destination "$envProgramFiles\YourApp" |

* **Copy-ADTFileToUserProfiles:** This function is specifically for copying files to each user profile on a system, supporting options like recursion and using Robocopy for long paths.

|  |
| --- |
| Copy-ADTFileToUserProfiles -Path "$dirSupportFiles\UserSpecificFile.ini" -Destination "AppData\Roaming\YourApp" |

* **Copy-Item:** The native PowerShell cmdlet can also be used for basic copy operations, especially when not needing the specific features of PSADT functions.

|  |
| --- |
| Copy-Item -Path "C:\Source\Folder" -Destination "C:\Destination\Folder" -Recurse |

**2. Deleting Files and Folders:**

* **Remove-File:** This PSADT function is used to delete specific files.

|  |
| --- |
| Remove-File -Path "$envUserDesktop\OldShortcut.lnk" |

* **Remove-Folder:** This PSADT function is used to delete entire folders.

|  |
| --- |
| Remove-Folder -Path "$envProgramFiles\OldApp" |

* **Remove-Item:** The native PowerShell cmdlet can also be used to delete files or folders. Use the -Force parameter to remove read-only or hidden items.

|  |
| --- |
| Remove-Item -Path "C:\Temp\OldLog.log" -Force |

**3. Moving Files and Folders:**

* Move-Item: This is the native PowerShell cmdlet for moving files and folders. PSADT does not provide a dedicated function for this, as Move-Item is generally sufficient.

|  |
| --- |
| Move-Item -Path "C:\Source\File.doc" -Destination "C:\Destination\Documents" |