

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

- In PSAppDeployToolkit, Execute-MSI is a function used for installing, uninstalling, patching, repairing, or performing active setup on MSI and MSP files, or using MSI product codes
- For installation, it defaults to using the /i (install) switch of msixec.exe
- For uninstallation, it uses the /x (uninstall)
- Executes msixec.exe to perform the following actions for MSI & MSP files and MSI product codes: install, uninstall, patch, repair, active setup.

Example of Execute-MSI :

1. Installs an MSI

Execute-MSI -Action 'Install' -Path 'Adobe_FlashPlayer_11.2.202.233_x64_EN.msi'

2. Installs an MSI, applying a transform and overriding the default MSI toolkit parameters

Execute-MSI -Action 'Install' -Path 'Adobe_FlashPlayer_11.2.202.233_x64_EN.msi' -Transform 'Adobe_FlashPlayer_11.2.202.233_x64_EN_01.mst' -Parameters '/QN'

3. Uninstalls an MSI using a product code

Execute-MSI -Action 'Uninstall' -Path '{26923b43-4d38-484f-9b9e-de460746276c}'

PARAMETERS :

1. -Action

The action to perform. Options: Install, Uninstall, Patch, Repair, ActiveSetup.

2. -Path

The path to the MSI/MSP file or the product code of the installed MSI.

3. -Transform

The name of the transform file(s) to be applied to the MSI. The transform file is expected to be in the same directory as the MSI file. Multiple transforms have to be separated by a semi-colon.

4. -Parameter

Overrides the default parameters specified in the XML configuration file

- **Execute-Process :**

- Execute a process with optional arguments, working directory, window style.

Example of Execute-Process :

1. If the file is in the "Files" directory of the App Deploy Toolkit, only the file name needs to be specified.

Execute-Process -Path 'uninstall_flash_player_64bit.exe' -Parameters '/uninstall' -WindowStyle 'Hidden'

2. `Execute-Process -Path "$dirFiles\Bin\setup.exe" -Parameters '/S' -WindowStyle 'Hidden'`
3. `Execute-Process -Path 'setup.exe' -Parameters '/S' -IgnoreExitCodes '1,2'`
4. **Launch InstallShield "setup.exe" from the ".\Files" sub-directory and force log files to the logging folder.**

`Execute-Process -Path 'setup.exe' -Parameters "-s -f2`"$configToolkitLogDir\$installName.log`""`

5. **Launch InstallShield "setup.exe" with embedded MSI and force log files to the logging folder**

`Execute-Process -Path 'setup.exe' -Parameters "/s /v`"ALLUSERS=1 /qn /L*`"$configToolkitLogDir\$installName.log`""`

PARAMETERS :

1. `-WindowStyle`
Style of the window of the process executed. Options: Normal, Hidden, Maximized, Minimized.
2. `-Parameters`
Arguments to be passed to the executable

- **Execute-MSP:**

Reads SummaryInfo targeted product codes in MSP file and determines if the MSP file applies to any installed products. If a valid installed product is found, triggers the `Execute-MSI` function to patch the installation. Uses default config MSI parameters. You can use `-AddParameters` to add additional parameters.

Examples of Execute-MSP :

1. `Execute-MSP -Path 'Adobe_Reader_11.0.3_EN.msp'`
2. `Execute-MSP -Path 'AcroRdr2017Upd1701130143_MUI.msp' -AddParameters 'ALLUSERS=1'`

PARAMETERS :

1. `-AddParameters`
Additional parameters
2. `-Path`
Path to the msp file

- **Copy-File :**

- Copy a file or group of files to a destination path.

Examples of Copy-File :

1. **Copy File to Destination path :**

`Copy-File -Path "$dirSupportFiles\MyApp.ini" -Destination "$envWinDir\MyApp.ini"`

2. Copy all of the files in a folder to a destination folder.

Copy-File -Path "\$dirSupportFiles*.*" -Destination "\$envTemp\tempfiles"

PARAMETERS :

1. -Path
Path of the file to copy. Multiple paths can be specified
2. -Destination
Destination Path of the file to copy.

• Install-MSUpdate :

- Install all Microsoft Updates of type ".exe", ".msu", or ".msp" in a given directory
- **Examples of Intall-MSUpdate :**
 1. Install-MSUpdates -Directory "\$dirFiles\MSUpdates"

PARAMETERS :

1. -Directory
Directory containing the updates.

• Set-ActiveSetup :

- Creates an Active Setup entry in the registry to execute a file for each user upon login.
- **Examples of Set-ActiveSetup :**
 1. Set-ActiveSetup -StubExePath 'C:\Users\Public\Company\ProgramUserConfig.vbs' -Arguments '/Silent' -Description 'Program User Config' -Key 'ProgramUserConfig' -Locale 'en'
 2. Set-ActiveSetup -StubExePath "\$envWinDir\regedit.exe" -Arguments "/S
`"%SystemDrive%\Program Files (x86)\PS App
Deploy\PSAppDeployHKCUSettings.reg`"" -Description 'PS App Deploy Config' -Key
'PS_App_Deploy_Config' -ContinueOnError \$true
 3. **Deletes "ProgramUserConfig" active setup entry from all registry hives.**
Set-ActiveSetup -Key 'ProgramUserConfig' -PurgeActiveSetupKey

PARAMETERS :

1. -Stub-Exe-Path
Use this parameter to specify the destination path of the file that will be executed upon user login.
2. -Arguments
Arguments to pass to the file being executed.
3. -Key
Name of the registry key for the Active Setup entry. Default is: \$installName.