

Execute-MSI and Uninstall-ADTApplication

Installation:

- **Execute-MSI:** This cmdlet is used to execute MSI files.

You can specify the path to the MSI file and various parameters like -Action Install, -Path, and -DeployMode (Interactive, Silent, NonInteractive).

- Example: `Execute-MSI -Action Install -Path "C:\path\to\your.msi" -DeployMode Silent`

Uninstallation (via PowerShell):

1. **Uninstall-ADTApplication**
→ Uninstalls apps installed with **PSAppDeployToolkit**.
2. **Execute-MSI -Action Uninstall -Path "{GUID}"**
→ Uses product **GUID** to uninstall via MSI.
3. **Start-Process with msixec.exe**
→ Directly calls Windows Installer to uninstall.
→ Example:

`Start-Process -FilePath "msiexec.exe" -ArgumentList "/x{GUID} /qn" -Wait`

All methods support **silent uninstall** using `/qn`.

Use **product GUID** to target the specific installed app.

Other related commands:

- **Get-RunningProcesses:** Checks for running processes that might need to be closed before an installation or uninstallation
- **Show-WelcomePrompt:** Displays a welcome prompt to the user.
- **Remove-MSIApplications:** Removes specified MSI applications.
- **Unblock-ADTAppExecution:** Unblocks the execution of an application that was blocked.
- **Unregister-ADTDII :** Unregisters a specific DLL.

Execute-MSI in PSAppDeployToolkit

- **Purpose:-** Used to install, uninstall, patch, or repair MSI/MSP files using msiexec.exe switches.

➤ Installation:

- ❖ **Basic Command:** `Execute-MSI -Action Install -Path "C:\path\to\app.msi"`

Common Parameters:

- ❖ `/qn` – Silent install
- ❖ `REBOOT=ReallySuppress` – Prevent reboot
- ❖ `/L*v` – Verbose logging

Example: Execute-MSI -Action Install -Path "app.msi" -Parameters "/qn REBOOT=ReallySuppress"

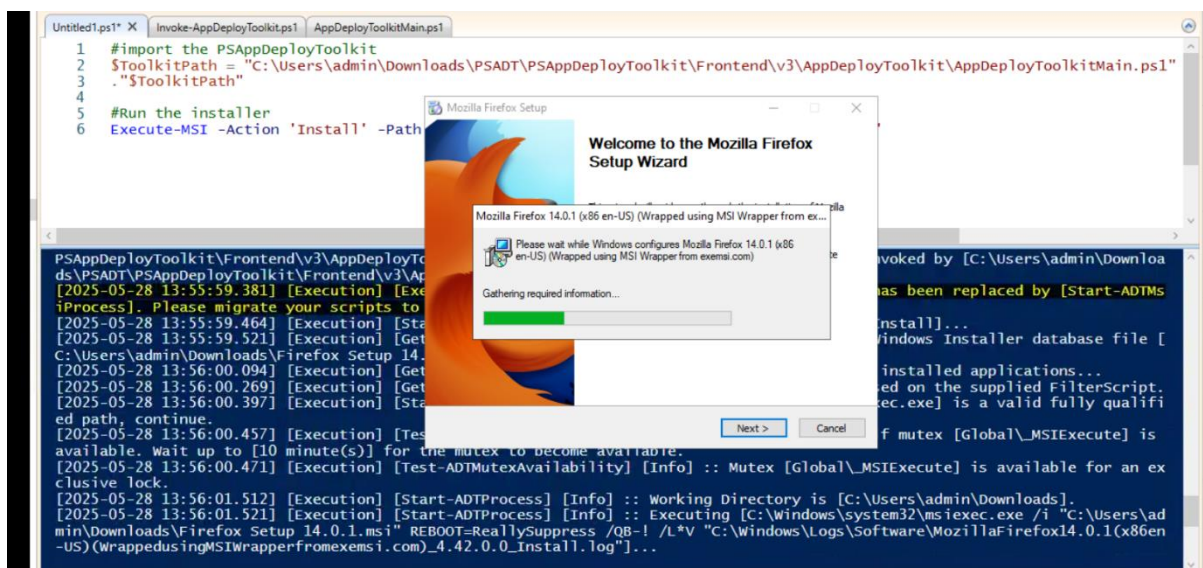
➤ Uninstallation:

- ❖ Basic Command:
- ❖ Execute-MSI -Action Uninstall -Path "C:\path\to\app.msi"
- ❖ Alternative: Use product GUID instead of MSI path.
- ❖ Same parameters apply (/qn, REBOOT=ReallySuppress, /L*v).

Important Options:

- Product Code Support: Can uninstall using product GUID.
- Log Output: Use -LogName to save logs.
- Skip Reinstall Check: Use -SkipMSIAAlreadyInstalledCheck.
- Error Control: Use -ContinueOnError to allow script to continue after failure.

Practical Implementation of the installation command:



```
1 #import the PSAppDeployToolkit
2 $ToolkitPath = "C:\Users\admin\Downloads\PSADT\PSAppDeployToolkit\Frontend\v3\AppDeployToolkit\AppDeployToolkitMain.ps1"
3 ".$ToolkitPath"
4
5 #Run the installer
6 Execute-MSI -Action 'Install' -Path 'C:\Users\admin\Downloads\Firefox Setup 14.0.1.msi'
```

Execute, Copy, Install and Active Setup Commands

List of commands:

1) Execute-MSI

Executes msixec.exe to perform the following actions for MSI & MSP files and we can install, uninstall, update etc as per the parameter -Action.

- Execute-MSI -Action 'Install' -Path 'Adobe_FlashPlayer_11.2.202.233_x64_EN.msi'
- 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'
- Execute-MSI -Action 'Uninstall' -Path '{26923b43-4d38-484f-9b9e-de460746276c}'
- Execute-MSI -Action 'Patch' -Path 'Adobe_Reader_11.0.3_EN.msp'
- Execute-MSI -Action Install -Path \$AppMSIName -SkipMSIAAlreadyInstalledCheck -ContinueOnError \$False -LogName "\${AppMSIName}_MSI"

2) Execute-Process

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

- Execute-Process -Path 'uninstall_flash_player_64bit.exe' -Parameters '/uninstall' -WindowStyle 'Hidden'
- Execute-Process -Path "\$dirFiles\Bin\setup.exe" -Parameters '/S' -WindowStyle 'Hidden'
- Execute-Process -Path 'setup.exe' -Parameters '/S' -IgnoreExitCodes '1,2'
- Execute-Process -Path 'setup.exe' -Parameters "-s -f2`"\$configToolkitLogDir\\$installName.log`""
- Execute-Process -Path 'setup.exe' -Parameters "/s /v`"ALLUSERS=1 /qn /L*`"\$configToolkitLogDir\\$installName.log`""

3) Execute-MSP

Reads SummaryInfo targeted product codes in MSP file and determines if the MSP file applies to any installed products.

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

4) Copy-File

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

- Copy-File -Path "\$dirSupportFiles*.*)" -Destination "\$envTemp\tempfiles"
- Copy-File -Path "\$dirSupportFiles\MyApp.ini" -Destination "\$envWinDir\MyApp.ini"

5) Install-MSUpdate

Install all Microsoft Updates of type ".exe", ".msu", or ".msp" in a given directory.

- Install-MSUpdates -Directory "\$dirFiles\MSUpdates"

6) Set-ActiveSetup

Creates an Active Setup entry in the registry to execute a file for each user upon login.

- Set-ActiveSetup -StubExePath 'C:\Users\Public\Company\ProgramUserConfig.vbs' -Arguments '/Silent' -Description 'Program User Config' -Key 'ProgramUserConfig' -Locale 'en'
- 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