* **BATCH ID: WiproNGA\_DWS\_B5\_25VID2550**
* **NAME:** Shrawani Shyam Balwadkar
* **RPS USER ID:** 34932
* **DATE**: 12-08-2025

**TOPIC:** **How to execute a process in PowerShell**

**Here is how to execute a process in PowerShell:**

**1. Using the Direct Execution Method:**

* **Navigate to the directory containing the executable file using the cd command.**
* **Execute the file by typing its name, preceded by `. \`. For example: .\YourFile.exe**
* **Alternatively, provide the full path to the file: C:\path\to\YourFile.exe [**[**1**](https://www.wikihow.com/Powershell-How-to-Run-Exe#:~:text=Start%2DProcess%20%2DFilePath%20%22filename.exe%22%20.%20This%20command%20will,file%2C%20e.g.%2C%20Start%2DProcess%20%2DFilePath%20%22C:%5Cpath%20to%5Cfilename.exe%22%20.)**]**

**2. Using the Start-Process cmdlet:**

* **This cmdlet allows you to run processes asynchronously, meaning PowerShell will continue executing the script without waiting for the process to finish.**
* **To use it, type Start-Process -FilePath "path\to\your\file.exe".**
* **You can also specify options such as:**
* **-WindowStyle: To control the window style (e.g., Minimized, Hidden).**
* **-ArgumentList: To pass command-line arguments to the process.**
* **-Credential: To run the process under different credentials.**

**3. Running PowerShell Scripts:**

* **Navigate to the directory containing the script.**
* **Execute the script using `. \` syntax: .\YourScript.ps1.**
* **To run a script with administrative privileges, open PowerShell as an administrator.**

**4. Running Commands in the Background:**

* **To run a process in the background, you can use the Start-Process cmdlet without the -Wait parameter.**
* **You can also end a line with an ampersand (&).**

**5. Using Invoke-Expression:**

* **The Invoke-Expression cmdlet allows you to execute expressions directly in your PowerShell session.**
* **This can be useful for evaluating and executing scripts or strings on the fly. [**[**2**](https://syncromsp.com/blog/powershell-run-exe/#:~:text=The%20Invoke%2DExpression%20command%20lets%20you%20execute%20expressions,multiple%20statements%20at%20once%2C%20which%20is%20especially)**]**

**6. Running Processes with Specific User Accounts:**

* **The Execute-ProcessAsUser function from the PSAppDeployToolkit can be used to run processes under a specific user account.**

**7. Changing the Execution Policy:**

* **To run scripts, you may need to adjust the PowerShell execution policy using the Set-ExecutionPolicy cmdlet.**

**Important Considerations:**

* **When executing processes, ensure you have the necessary permissions.**
* **Be cautious when running scripts or executables from untrusted sources.**
* **Use the -Wait parameter with Start-Process if you need PowerShell to wait for the process to finish before continuing.**

**Active Setup, when integrated with the PowerShell App Deployment Toolkit (PSADT), provides a mechanism to execute per-user changes or configurations upon user login. This is particularly useful for applications that require user-specific settings or file modifications, especially when the application is installed by a privileged account (e.g., SYSTEM) and lacks advertised entry points like shortcuts that would trigger a repair for the current user.**

**How it works with PSADT:**

* **Registry Entries: PSADT's Set-ActiveSetup function creates specific registry entries in HKLM:\SOFTWARE\Microsoft\Active Setup\Installed Components\<ProductCode>. These entries define the executable path (StubExePath), arguments, and a version number.**
* **User Login Trigger: When a user logs in, the operating system compares the Active Setup entries in HKLM with those in the user's HKCU hive.**
* **Execution Condition: If the HKCU entry is missing or its version number is lower than the HKLM entry, the executable specified in StubExePath is run for that user.**
* **Version Management: PSADT automatically handles the versioning, typically using a granular timestamp (YYYYMMDDHHMMSS) to ensure Active Setup is triggered even for re-installations on the same day.**
* **Per-User Execution: The Set-ActiveSetup function also includes parameters to control whether the StubExePath is executed for the current user immediately during installation, or only upon subsequent user logins.**
* **Cleanup: The -PurgeActiveSetupKey parameter within Set-ActiveSetup can be used during uninstallation to remove the Active Setup entry from all loaded HKCU hives, ensuring a clean removal. [**[**1**](https://psappdeploytoolkit.com/docs/reference/functions/Set-ADTActiveSetup)**]**

**Key Benefits of using PSADT for Active Setup:**

* **Standardization: PSADT provides a structured framework for implementing Active Setup, ensuring consistency across deployments.**
* **Automation: The Set-ActiveSetup function automates the creation and management of Active Setup registry entries.**
* **Flexibility: Parameters within Set-ActiveSetup allow for customization of the executable path, arguments, description, and versioning.**
* **Cleanup Capabilities: The toolkit facilitates the removal of Active Setup entries during uninstallation.**