

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]

## 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]

## 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.