Start-Process

Module: Microsoft.PowerShell.Management

Starts one or more processes on the local computer.

In this article

PowerShell Copy

```
Start-Process
     [-FilePath] <String>
     [[-ArgumentList] <String[]>]
     [-Credential <PSCredential>]
     [-WorkingDirectory <String>]
     [-LoadUserProfile]
     [-NoNewWindow]
     [-PassThru]
     [-RedirectStandardError <String>]
     [-RedirectStandardInput <String>]
     [-RedirectStandardOutput <String>]
     [-WindowStyle <ProcessWindowStyle>]
     [-Wait]
     [-UseNewEnvironment]
     [<CommonParameters>]
```

PowerShell Copy

```
Start-Process
     [-FilePath] <String>
     [[-ArgumentList] <String[]>]
     [-WorkingDirectory <String>]
     [-PassThru]
     [-Verb <String>]
     [-WindowStyle <ProcessWindowStyle>]
     [-Wait]
     [<CommonParameters>]
```

Description

The **Start-Process** cmdlet starts one or more processes on the local computer. To specify the program that runs in the process, enter an executable file or script file, or a file that can be opened by using a program on the computer. If you specify a non-executable file, **Start-Process** starts the program that is associated with the file, similar to the Invoke-Item cmdlet.

You can use the parameters of **Start-Process** to specify options, such as loading a user profile, starting the process in a new window, or using alternate credentials.

Examples

Example 1: Start a process that uses default values

PowerShell Copy

```
PS C:\> Start-Process -FilePath "sort.exe"
```

This command starts a process that uses the Sort.exe file in the current folder. The command uses all of the default values, including the default window style, working folder, and credentials.

Example 2: Print a text file

PowerShell Copy

```
PS C:\> Start-Process -FilePath "myfile.txt" -WorkingDirectory "C:\PS-Test" -Verb Print
```

This command starts a process that prints the C:\PS-Test\MyFile.txt file.

Example 3: Start a process to sort items to a new file

PowerShell Copy

```
PS C:\> Start-Process -FilePath "Sort.exe" -RedirectStandardInput "Testsort.txt" -
RedirectStandardOutput "Sorted.txt" -RedirectStandardError "SortError.txt" -
UseNewEnvironment
```

This command starts a process that sorts items in the Testsort.txt file and returns the sorted items in the Sorted.txt files. Any errors are written to the SortError.txt file.

The *UseNewEnvironment* parameter specifies that the process runs with its own environment variables.

Example 4: Start a process in a maximized window

PowerShell Copy

```
PS C:\> Start-Process -FilePath "notepad" -Wait -WindowStyle Maximized
```

This command starts the Notepad process. It maximizes the window and retains the window until the process completes.

Example 5: Start Windows Powershell as an administrator

PowerShell Copy

```
PS C:\> Start-Process -FilePath "powershell" -Verb runAs
```

This command starts Windows PowerShell by using the Run as administrator option.

Example 6: Using different verbs to start a process

PowerShell Copy

```
PS C:\> $startExe = New-Object System.Diagnostics.ProcessStartInfo -Args PowerShell.exe
PS C:\> $startExe.verbs
open
runas

# Starts a PowerShell process in a new console window.

PS C:\> Start-Process -FilePath "powershell.exe" -Verb open

# Starts a PowerShell process with "Run as Administrator" permissions.

PS C:\> Start-Process -FilePath "powershell.exe" -Verb runas
```

These commands show how to find the verbs that can be used when starting a process, and the effect of using the verbs to start the process.

The available verbs are determined by the file name extension of the file that runs in the process. To find the verbs for a process, create a **System.Diagnostics.ProcessStartInfo** object for the process file and look in the **Verbs** property of the object. This example uses the PowerShell.exe file that runs in the PowerShell process.

The first command uses **New-Object** to create a **System.Diagnostics.ProcessStartInfo** object for PowerShell.exe, the file that runs in the PowerShell process. The command saves the **ProcessStartInfo** object in the $startExe variable.

The second command displays the values in the **Verbs** property of the **ProcessStartInfo** object in the $startExe variable. The results show that you can use the Open and Runas verbs with PowerShell.exe, or with any process that runs a .exe file.

The third command starts a PowerShell process with the Open verb. The Open verb starts the process in a new console window.

The fourth command starts a PowerShell process with the RunAs verb. The RunAs verb starts the process with permissions of a member of the Administrators group on the computer. This is the same as starting Windows PowerShell by using the Run as administrator option.

Example 7: Specifying arguments to the process

PowerShell Copy

```
PS C:\> Start-Process -FilePath "$env:comspec" -ArgumentList "/c dir
`"%systemdrive%\program files`""
PS C:\> Start-Process -FilePath "$env:comspec" -ArgumentList
"/c","dir","`"%systemdrive%\program files`""
```

Both commands start the Windows command interpreter, issueing a dir command on the 'Program Files' folder. Because this foldername contains a space, the value needs surrounded with escaped quotes. Note that the first command specifies a string as ArgumentList. The second command a string array.

Required Parameters

`-FilePath`

Specifies the optional path and file name of the program that runs in the process. Enter the name of an executable file or of a document, such as a .txt or .doc file, that is associated with a program on the computer. This parameter is required.

If you specify only a file name, use the *WorkingDirectory* parameter to specify the path.

Type:
String

Aliases:
PSPath

Position:

0

Default value:
None

Accept pipeline input:
False

Accept wildcard characters:
False

Optional Parameters

`-ArgumentList`

Specifies parameters or parameter values to use when this cmdlet starts the process. If parameters or parameter values contain a space, they need surrounded with escaped double quotes.

Type:
String[]

Aliases:
Args

Position:
1

Default value:
None

Accept pipeline input:
False

Accept wildcard characters:
False

`-Credential`

Specifies a user account that has permission to perform this action. Type a user name, such as User01 or Domain01\User01, or enter a **PSCredential** object, such as one from the Get-Credential cmdlet. By default, the cmdlet uses the credentials of the current user.

Type:

PSCredential

Aliases:
RunAs

Position:
Named

Default value:
None

Accept pipeline input:
False

Accept wildcard characters:
False

`-LoadUserProfile`

Indicates that this cmdlet loads the Windows user profile stored in the **HKEY_USERS** registry key for the current user.

This parameter does not affect the Windows PowerShell profiles. For more information, see about_Profiles.

Type:
SwitchParameter

Aliases:
Lup

Position:
Named

Default value:
None

Accept pipeline input:
False

Accept wildcard characters:
False

`-NoNewWindow`

Start the new process in the current console window. By default Windows PowerShell opens a new window.

You cannot use the *NoNewWindow* and *WindowStyle* parameters in the same command.

Type:
SwitchParameter

Aliases:
nnw

Position:
Named

Default value:
None

Accept pipeline input:
False

Accept wildcard characters:
False

`-PassThru`

Returns a process object for each process that the cmdlet started. By default, this cmdlet does not generate any output.

Type:
SwitchParameter

Position:
Named

Default value:
None

Accept pipeline input:
False

Accept wildcard characters:
False

`-RedirectStandardError`

Specifies a file. This cmdlet sends any errors generated by the process to a file that you specify. Enter the path and file name. By default, the errors are displayed in the console.

Type:
String

Aliases:
RSE

Position:
Named

Default value:
None

Accept pipeline input:
False

Accept wildcard characters:
False

`-RedirectStandardInput`

Specifies a file. This cmdlet reads input from the specified file. Enter the path and file name of the input file. By default, the process gets its input from the keyboard.

Type:
String

Aliases:
RSI

Position:
Named

Default value:
None

Accept pipeline input:
False

Accept wildcard characters:
False

`-RedirectStandardOutput`

Specifies a file. This cmdlet sends the output generated by the process to a file that you specify. Enter the path and file name. By default, the output is displayed in the console.

Type:
String

Aliases:
RSO

Position:
Named

Default value:
None

Accept pipeline input:
False

Accept wildcard characters:
False

`-UseNewEnvironment`

Indicates that this cmdlet uses new environment variables specified for the process. By default, the started process runs with the environment variables specified for the computer and user.

Type:
SwitchParameter

Position:
Named

Default value:
None

Accept pipeline input:
False

Accept wildcard characters:
False

`-Verb`

Specifies a verb to use when this cmdlet starts the process. The verbs that are available are determined by the file name extension of the file that runs in the process.

The following table shows the verbs for some common process file types.

File type	Verbs
.cmd	Edit, Open, Print, Runas
.exe	Open, RunAs
.txt	Open, Print, PrintTo
.wav	Open, Play

To find the verbs that can be used with the file that runs in a process, use the New-Object cmdlet to create a **System.Diagnostics.ProcessStartInfo** object for the file. The available verbs are in the **Verbs** property of the **ProcessStartInfo** object. For details, see the examples.

Type:
String

Position:
Named

Default value:
None

Accept pipeline input:
False

Accept wildcard characters:
False

`-Wait`

Indicates that this cmdlet waits for the specified process to complete before accepting more input. This parameter suppresses the command prompt or retains the window until the process finishes.

Type:
SwitchParameter

Position:

Named

Default value:
None

Accept pipeline input:
False

Accept wildcard characters:
False

`-WindowStyle`

Specifies the state of the window that is used for the new process. The acceptable values for this parameter are: Normal, Hidden, Minimized, and Maximized. The default value is Normal.

You cannot use the *WindowStyle* and *NoNewWindow* parameters in the same command.

Type:
ProcessWindowStyle

Accepted values:
Normal, Hidden, Minimized, Maximized

Position:
Named

Default value:
None

Accept pipeline input:
False

Accept wildcard characters:
False

`-WorkingDirectory`

Specifies the location of the executable file or document that runs in the process. The default is the folder for the new process.

Type:
String

Position:

Named

Default value:
None

Accept pipeline input:
False

Accept wildcard characters:
False

Inputs

None

You cannot pipe input to this cmdlet.

Outputs

None, System.Diagnostics.Process

This cmdlet generates a **System.Diagnostics.Process** object, if you specify the *PassThru* parameter. Otherwise, this cmdlet does not return any output.

Notes

- This cmdlet is implemented by using the **Start** method of the **System.Diagnostics.Process** class. For more information about this method, see Process.Start Method in the MSDN library.

Related Links

- Debug-Process
- Get-Process
- Start-Service
- Stop-Process
- Wait-Process