**Working With Files, Folders and Registry Keys**

* Article
* 10/08/2021
* 4 minutes to read
* 4 contributors

Windows PowerShell uses the noun **Item** to refer to items found on a Windows PowerShell drive. When dealing with the Windows PowerShell FileSystem provider, an **Item** might be a file, a folder, or the Windows PowerShell drive. Listing and working with these items is a critical basic task in most administrative settings, so we want to discuss these tasks in detail.

**Enumerating Files, Folders, and Registry Keys (Get-ChildItem)**

Since getting a collection of items from a particular location is such a common task, the Get-ChildItem cmdlet is designed specifically to return all items found within a container such as a folder.

If you want to return all files and folders that are contained directly within the folder C:\Windows, type:

Copy

PS> Get-ChildItem -Path C:\Windows

Directory: Microsoft.Windows PowerShell.Core\FileSystem::C:\Windows

Mode LastWriteTime Length Name

---- ------------- ------ ----

-a--- 2006-05-16 8:10 AM 0 0.log

-a--- 2005-11-29 3:16 PM 97 acc1.txt

-a--- 2005-10-23 11:21 PM 3848 actsetup.log

...

The listing looks similar to what you would see when you enter the dir command in **Cmd.exe**, or the ls command in a UNIX command shell.

You can perform very complex listings by using parameters of the Get-ChildItem cmdlet. We will look at a few scenarios next. You can see the syntax the Get-ChildItem cmdlet by typing:

PowerShellCopy

Get-Command -Name Get-ChildItem -Syntax

These parameters can be mixed and matched to get highly customized output.

**Listing all Contained Items (-Recurse)**

To see both the items inside a Windows folder and any items that are contained within the subfolders, use the **Recurse** parameter of Get-ChildItem. The listing displays everything within the Windows folder and the items in its subfolders. For example:

Copy

PS> Get-ChildItem -Path C:\WINDOWS -Recurse

Directory: Microsoft.Windows PowerShell.Core\FileSystem::C:\WINDOWS

Directory: Microsoft.Windows PowerShell.Core\FileSystem::C:\WINDOWS\AppPatch

Mode LastWriteTime Length Name

---- ------------- ------ ----

-a--- 2004-08-04 8:00 AM 1852416 AcGenral.dll

...

**Filtering Items by Name (-Name)**

To display only the names of items, use the **Name** parameter of Get-Childitem:

Copy

PS> Get-ChildItem -Path C:\WINDOWS -Name

addins

AppPatch

assembly

...

**Forcibly Listing Hidden Items (-Force)**

Items that are normally invisible in File Explorer or Cmd.exe are not displayed in the output of a Get-ChildItem command. To display hidden items, use the **Force** parameter of Get-ChildItem. For example:

PowerShellCopy

Get-ChildItem -Path C:\Windows -Force

This parameter is named Force because you can forcibly override the normal behavior of the Get-ChildItem command. Force is a widely used parameter that forces an action that a cmdlet would not normally perform, although it will not perform any action that compromises the security of the system.

**Matching Item Names with Wildcards**

The Get-ChildItem command accepts wildcards in the path of the items to list.

Because wildcard matching is handled by the Windows PowerShell engine, all cmdlets that accepts wildcards use the same notation and have the same matching behavior. The Windows PowerShell wildcard notation includes:

* Asterisk (\*) matches zero or more occurrences of any character.
* Question mark (?) matches exactly one character.
* Left bracket ([) character and right bracket (]) character surround a set of characters to be matched.

Here are some examples of how wildcard specification works.

To find all files in the Windows directory with the suffix .log and exactly five characters in the base name, enter the following command:

Copy

PS> Get-ChildItem -Path C:\Windows\?????.log

Directory: Microsoft.Windows PowerShell.Core\FileSystem::C:\Windows

Mode LastWriteTime Length Name

---- ------------- ------ ----

...

-a--- 2006-05-11 6:31 PM 204276 ocgen.log

-a--- 2006-05-11 6:31 PM 22365 ocmsn.log

...

-a--- 2005-11-11 4:55 AM 64 setup.log

-a--- 2005-12-15 2:24 PM 17719 VxSDM.log

...

To find all files that begin with the letter x in the Windows directory, type:

PowerShellCopy

Get-ChildItem -Path C:\Windows\x\*

To find all files whose names begin with "x" or "z", type:

PowerShellCopy

Get-ChildItem -Path C:\Windows\[xz]\*

For more information about wildcards, see [about\_Wildcards](https://docs.microsoft.com/en-us/powershell/module/microsoft.powershell.core/about/about_wildcards).

**Excluding Items (-Exclude)**

You can exclude specific items by using the **Exclude** parameter of Get-ChildItem. This lets you perform complex filtering in a single statement.

For example, suppose you are trying to find the Windows Time Service DLL in the **System32** folder, and all you can remember about the DLL name is that it begins with "W" and has "32" in it.

An expression like w\*32\*.dll will find all DLLs that satisfy the conditions, but you may want to further filter out the files and omit any win32 files. You can omit these files by using the **Exclude** parameter with the pattern win\*:

Copy

PS> Get-ChildItem -Path C:\WINDOWS\System32\w\*32\*.dll -Exclude win\*

Directory: C:\WINDOWS\System32

Mode LastWriteTime Length Name

---- ------------- ------ ----

-a--- 3/18/2019 9:43 PM 495616 w32time.dll

-a--- 3/18/2019 9:44 PM 35328 w32topl.dll

-a--- 1/24/2020 5:44 PM 401920 Wldap32.dll

-a--- 10/10/2019 5:40 PM 442704 ws2\_32.dll

-a--- 3/18/2019 9:44 PM 66048 wsnmp32.dll

-a--- 3/18/2019 9:44 PM 18944 wsock32.dll

-a--- 3/18/2019 9:44 PM 64792 wtsapi32.dll

**Mixing Get-ChildItem Parameters**

You can use several of the parameters of the Get-ChildItem cmdlet in the same command. Before you mix parameters, be sure that you understand wildcard matching. For example, the following command returns no results:

PowerShellCopy

Get-ChildItem -Path C:\Windows\\*.dll -Recurse -Exclude [a-y]\*.dll

There are no results, even though there are two DLLs that begin with the letter "z" in the Windows folder.

No results were returned because we specified the wildcard as part of the path. Even though the command was recursive, the Get-ChildItem cmdlet restricted the items to those that are in the Windows folder with names ending with .dll.

To specify a recursive search for files whose names match a special pattern, use the **Include** parameter.

Copy

PS> Get-ChildItem -Path C:\Windows -Include \*.dll -Recurse -Exclude [a-y]\*.dll

Directory: Microsoft.Windows PowerShell.Core\FileSystem::C:\Windows\System32\Setup

Mode LastWriteTime Length Name

---- ------------- ------ ----

-a--- 2004-08-04 8:00 AM 8261 zoneoc.dll

Directory: Microsoft.Windows PowerShell.Core\FileSystem::C:\Windows\System32

Mode LastWriteTime Length Name

---- ------------- ------ ----

-a--- 2004-08-04 8:00 AM 337920 zipfldr.dll