How to Create a Folder in PowerShell

lazyadmin.nl/powershell/create-folder

July 9, 2023

Creating new folders from command line tools has always been possible in Windows. We could use the command `mkdir` in the old command prompt. But how do you create a folder in PowerShell? You could use the command mkdir as well, but there is a better way.

In PowerShell, we can use the cmdlet New-Item to create a new directory. This cmdlet is more powerful than the old mkdir command.

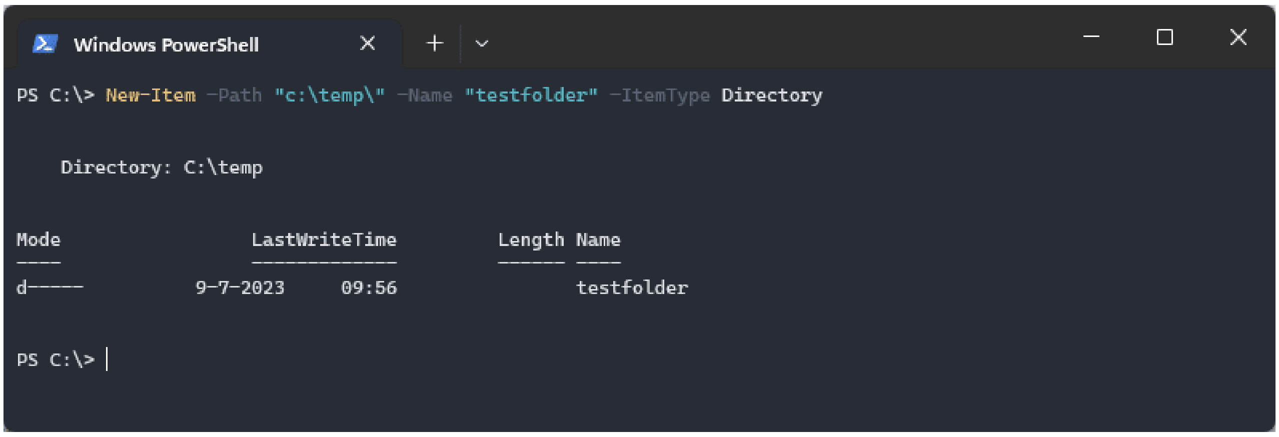
In this article, I will show you how you can create a folder in PowerShell, with or without subdirectories, and more!

Powershell Create Folder

Let's start with the basics, to create a new directory in PowerShell we are going to use the cmdlet `New-Item`. This cmdlet can be used to create files, folders, or symbolic links. In this article, we will focus on creating a folder.

To create a new directory we will need to specify the path where we want to create the folder and the name of the new folder:

New-Item -Path "c:\temp\" -Name "testfolder" -ItemType Directory



```
PS C:\> New-Item -Path "c:\temp\" -Name "testfolder" -ItemType Directory

    Directory: C:\temp

Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----          9-7-2023     09:56                testfolder

PS C:\> |
```

We can even shorten this by including the new folder name in the `-path` parameter and replacing `-ItemType` to `-Type`

# Create a new folder "testfolder-b" in the path c:\temp
New-Item -Path "c:\temp\testfolder-b" -Type Directory
# You can even leave out the path paremeter label
New-Item "c:\temp\testfolder-b" -Type Directory
Personally, I prefer to use the `-Name` parameter in scripts, because it makes it easier for others to read your code and see which new folder is created where.

Create new Directory in current Location

To make a new directory in the current location of the PowerShell script (or the working location of your PowerShell session), we can simply leave out the `-path` parameter. Another, and more readable, option is to specify a `.` for the current path:

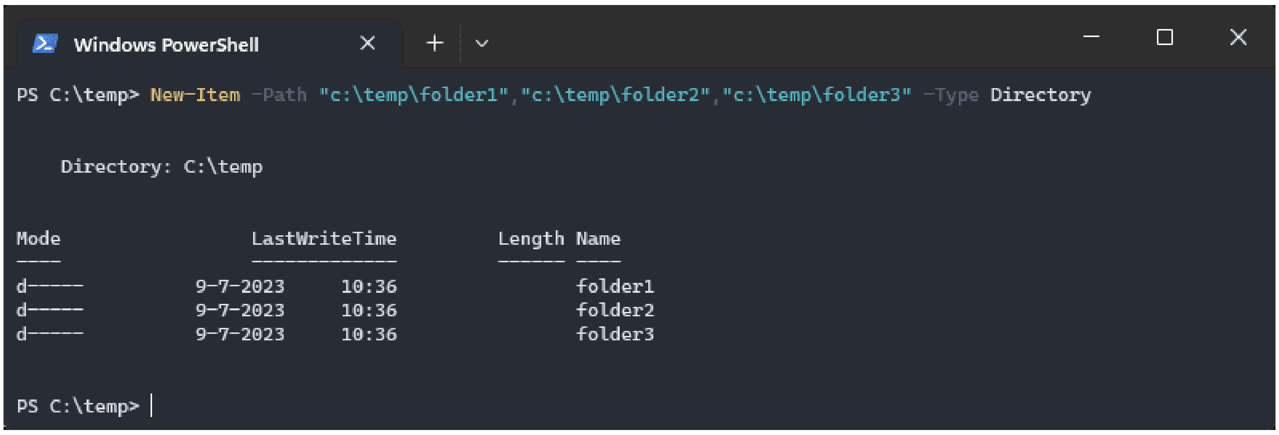
```
# Create a new directory in the current location
New-Item -Name "testfolder-c" -Type Directory
# Same result
New-Item -Path '.\testfolder-c' -Type Directory
```

Creating Multiple Folders

It's also possible to create multiple folders at once. To do this there are basically two methods that we can use. We can either loop through an array of folders to create or specify multiple folders in an array string.

If you only need to create a handful of folders, then the option below is the most convenient one to use:

```
New-Item -Path "c:\temp\folder1","c:\temp\folder2","c:\temp\folder3" -Type Directory
```



```
PS C:\temp> New-Item -Path "c:\temp\folder1","c:\temp\folder2","c:\temp\folder3" -Type Directory


    Directory: C:\temp


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----          9-7-2023     10:36                folder1
d-----          9-7-2023     10:36                folder2
d-----          9-7-2023     10:36                folder3


PS C:\temp>
```

The other option is to use a ForEach-Object loop. This method can be used with a simple array, or can be used to create folders from a CSV file for example:

```
$folders = (
"folder4",
"folder5",
"folder6"
)
$folders | ForEach {New-Item -Path "c:\temp\" -Name $_ -Type Directory}
```

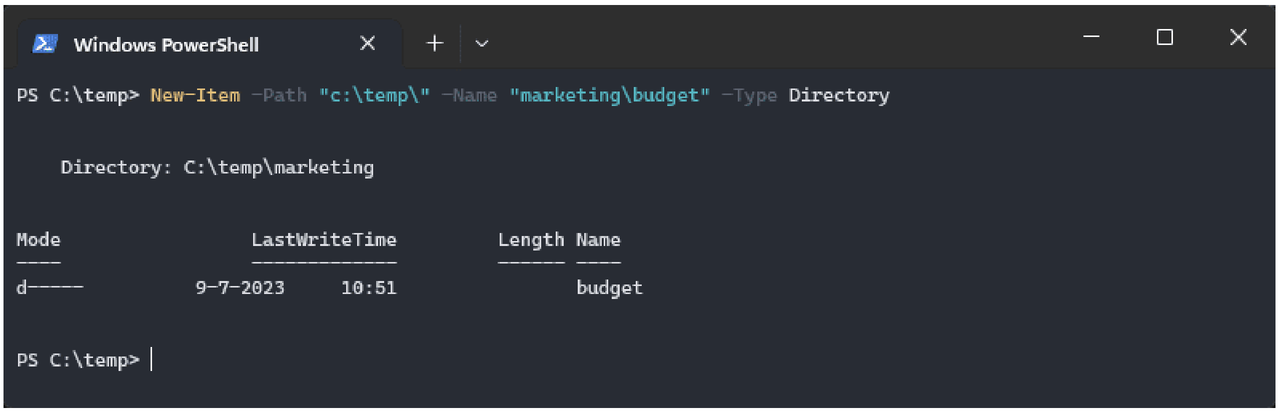
In the example below I have a simple CSV file with only the names of the folder. So we will need to add a header to be able to reference the folder names in the foreach loop. You can also pipe the ForEach loop directly behind the Import-CSV of course.

```
$folders = Import-CSV -Path C:\temp\example.csv -Header foldername
$folders | ForEach {New-Item -Path "c:\temp\" -Name $_.foldername -Type Directory}
```

Create Folders and Subfolders

One of the advantages of using PowerShell to create folders is that we can create folders and subfolders with a single command. This method is really handy when you need to create the same folder structure for new projects for example.

The `New-Item` cmdlet will create the parent folder automatically when it's missing. So for example, if we want to create the directory budget inside the folder marketing, then we only need to specify the name "marketing\budget". If the folder marketing doesn't exist, then the `New-Item` cmdlet will create it automatically.



```
PS C:\temp> New-Item -Path "c:\temp\" -Name "marketing\budget" -Type Directory


    Directory: C:\temp\marketing


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----          9-7-2023     10:51                budget


PS C:\temp>
```

In the example above, I have specified the new folders in the `-Name` parameter. But that isn't necessary. You can also specify the complete path in the `-Path` parameter or specify the parent path and only the subfolder in the `-Name` parameter. The commands below will all give the same result:

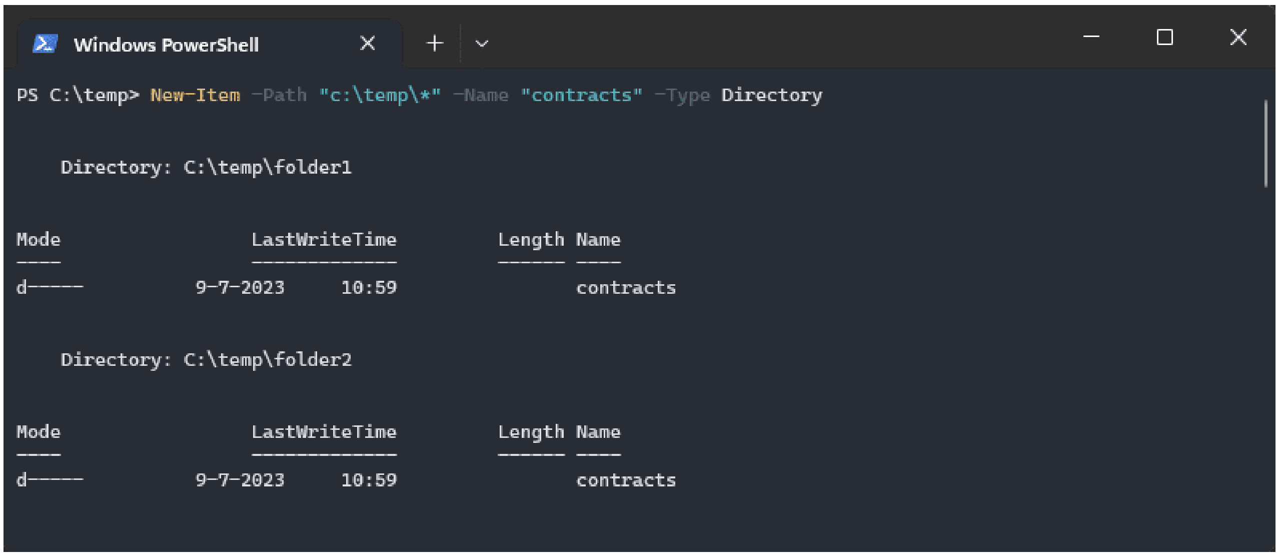
# All these commands will create the folder marketing and budget if they don't exist:
New-Item -Path "c:\temp\" -Name "marketing\budget" -Type Directory
New-Item -Path "c:\temp\marketing" -Name "budget" -Type Directory
New-Item -Path "c:\temp\marketing\budget" -Type Directory

Using Wildcards to create Subfolders

Another powerful feature of the new-item cmdlet is the ability to use a wildcard in the path parameter. Now you might be wondering why you would need this when you create a folder with PowerShell. But this method can really be handy when you need to add a subfolder to multiple folders.

For example, you have a folder for each of your clients. You are asked to create a new subfolder for each client named contracts. Instead of getting each directory first with the Get-ChildItem cmdlet, you can simply use a wildcard:

New-Item -Path "c:\temp\*" -Name "contracts" -Type Directory
As you can see, the subfolder contracts is created in each folder in the path c:\temp:



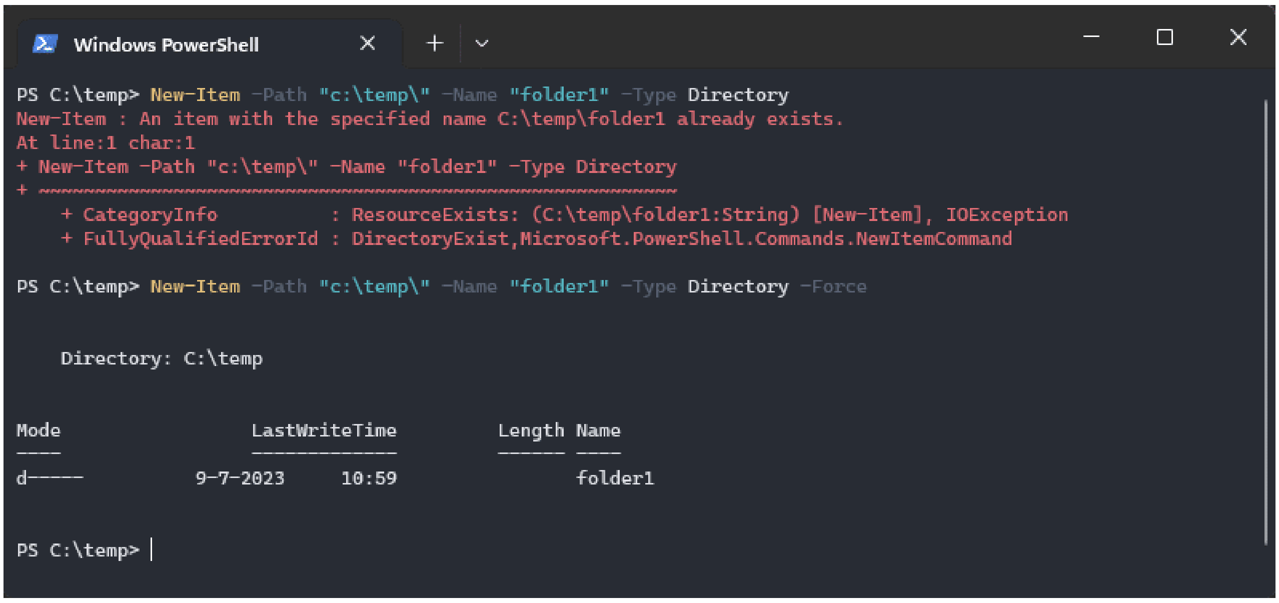
```
PS C:\temp> New-Item -Path "c:\temp\*" -Name "contracts" -Type Directory

    Directory: C:\temp\folder1

Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----          9-7-2023     10:59                contracts

    Directory: C:\temp\folder2

Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----          9-7-2023     10:59                contracts
```

Overwrite a Folder and using Test-Path

When a directory exists you can't create it again, there is even no need for it. But when creating folders from a script you often want it to continue even if the directory exists. There are a couple of options for that, the preferred method is to use the Test-Path cmdlet. But we can also force the creation of the folder or simply mute the error (which is the least preferred option).

The `-Force` parameter does not really recreate the folder, it will only return the existing folder object without showing any errors:

New-Item -Path "c:\temp\" -Name "folder1" -Type Directory -Force



```
PS C:\temp> New-Item -Path "c:\temp\" -Name "folder1" -Type Directory
New-Item : An item with the specified name C:\temp\folder1 already exists.
At line:1 char:1
+ New-Item -Path "c:\temp\" -Name "folder1" -Type Directory
+ ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    + CategoryInfo          : ResourceExists: (C:\temp\folder1:String) [New-Item], IOException
    + FullyQualifiedErrorId : DirectoryExist,Microsoft.PowerShell.Commands.NewItemCommand

PS C:\temp> New-Item -Path "c:\temp\" -Name "folder1" -Type Directory -Force

    Directory: C:\temp

Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----          9-7-2023     10:59                folder1

PS C:\temp> |
```

As mentioned it's better to use the `Test-Path` cmdlet to check if a folder exists before you create it. This allows you to create a better error-handling solution in your scripts. To create the directory only when the folder doesn't exist you can do the following:

```
$path = "C:\temp\folder1"
# Create the folder is the path doesn't exist
```

```
If (-not(test-path -Path $path)) {
New-Item -Path $path
}
```

Wrapping Up

The New-item cmdlet is a powerful cmdlet when it comes to creating files and folders in PowerShell. It replaces the mkdir command and allows you to easily create on or multiple folders including subfolders.

I hope you found this article helpful, if you have any questions, just drop a comment below.