How to Check if File Exists in PowerShell

lazyadmin.nl/powershell/check-if-file-exists

September 30, 2024

Before you try to read or write to a file in PowerShell, it’s important to check if the file actually exists. This will prevent errors, which can cause your script to fail. It’s also important to check before you create a new file, or when you try to delete a file.

In this article, we will look at the different methods of how you can check if a file exists and how to use it in your scripts.

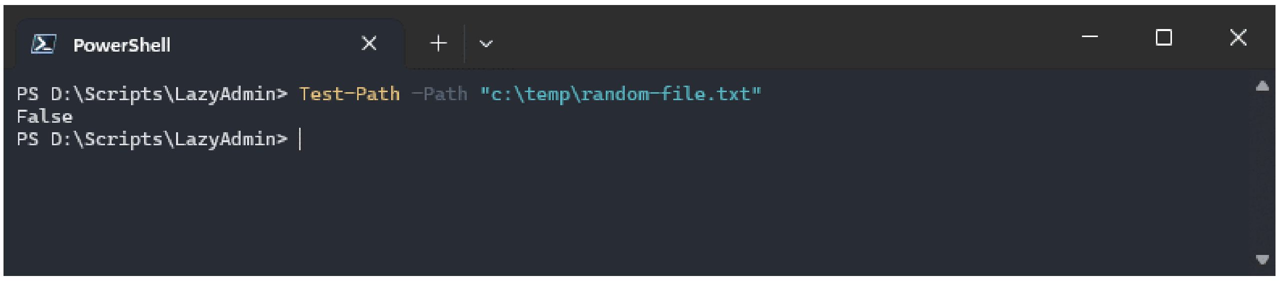
Check if a File Exists

There are basically two methods in PowerShell to check for the existence of a file. The recommended way is to use the Test-Path cmdlet, and another option is to use the Get-ChildItem cmdlet.

The best way to check if a file exists in PowerShell is to use the Test-Cmdlet. This cmdlet will return a boolean which can be `$True` or `$False`, making it easy to use inside an If-Statement for example.

If you run the cmdlet directly in your console, you will see that it will just return True or False as a string:

Test-Path -Path "c:\temp\random-file.txt"



```
PS D:\Scripts\LazyAdmin> Test-Path -Path "c:\temp\random-file.txt"
False
PS D:\Scripts\LazyAdmin> |
```

A more realistic example is to check if a file already exists before creating a new one. We will store the path in a variable, and use the `-Not` operator, to run the code only when the `Test-Path` result is false:

```
$path = "C:\temp\File1.txt"
if (-not(Test-Path -path $path)) {
# File doesn't exists, create the file
New-Item -Path $path
}
```

You can also use wildcards in the file path. This allows you to do a couple of things. First of all, if you don’t know the exact file name, then you can search for a part of the name.

It also allows you to check if there are any specific file types in a folder. For example, if you want to know if there is a log file or bak file in a folder.

```
# Check for specific file type
if (Test-Path -path "c:\temp\*.bak") {
Write-Host "Bak file(s) found in the folder"
}
# Check on part of the file name
if (Test-Path -path "c:\temp\la-*.txt") {
Write-Host "Found one or more files that start with la-"
}
```

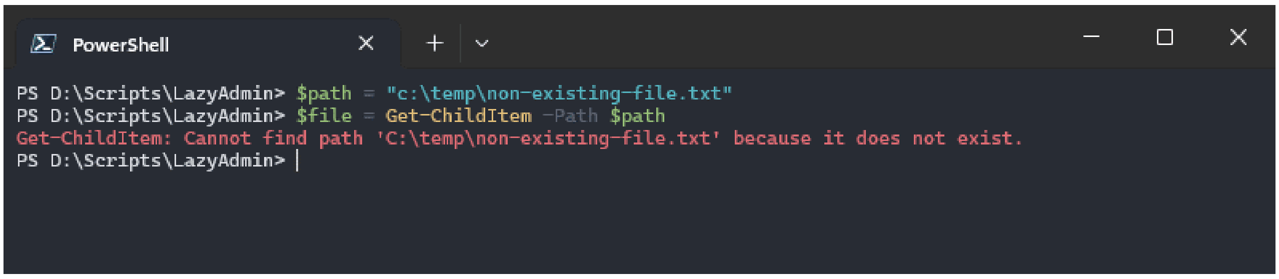
Using Get-ChildItem

Another method to check if a file exists in PowerShell is by using the Get-ChildItem cmdlet. This cmdlet is used for getting the specified items from a given directory. But we can also use the cmdlet to check if a file exists.

The difference between the `Get-ChildItem` and the `Test-Path` cmdlet is that the latter only returns `True` or `False`, whereas the first returns the file information as well.

When using the Get-ChildItem cmdlet, you can't just provide the complete path and file name. The reason for this is, that if the file doesn't exist, you will get an error, which will stop your script:

```
# This won't work if the file doesn't exist
$path = "c:\temp\non-existing-file.txt"
$file = Get-ChildItem -Path $path
if ($file) {
# Do something with the file information
}else{
Write-Host "File doesn't exist"
}
```



```
PS D:\Scripts\LazyAdmin> $path = "c:\temp\non-existing-file.txt"
PS D:\Scripts\LazyAdmin> $file = Get-ChildItem -Path $path
Get-ChildItem: Cannot find path 'C:\temp\non-existing-file.txt' because it does not exist.
PS D:\Scripts\LazyAdmin>
```

To use the `Get-ChildItem` cmdlet for this, we will need to provide the path and file name that we are looking for separately.

We use the `Filter` property to filter out the files in the path that match the file that we are looking for:

```
$path = "c:\temp\"
$fileName = "non-existing-file.txt"
# Get all the files in the path and filter on filename
$file = Get-ChildItem -Path $path -Filter $fileName
if ($file) {
# Do something with the file information
}else{
Write-Host "File doesn't exist"
}
```

Search in SubFolders

The advantage of the Get-ChildItem cmdlet is that we can also search in the subfolders as well for the file. We can add the parameter -recurse so it will search in all nested folders as well:

```
$path = "c:\temp\"
$fileName = "file1.txt"
$file = Get-ChildItem -Path $path -Filter $fileName -Recurse -File
if ($file) {
# Do something with the file information
Write-Host "File created on $($file.CreationTime)"
}
```

As you can see in the example above, I have added the parameter `-File` as well. In the examples I have searched on the filename including the extension, then it isn't really necessary. But when you search on only the filename, then directories can be included as well. The parameter -File will limit the result to only files.

Wrapping Up

If you want to check if a file exists in PowerShell, then the Test-Path cmdlet is the best option to use. It returns True or False depending on the result, making it the best option to use inside scripts.

When you don't know the exact location of the file or want to check if there are one or more files with a specific name, then the Get-ChildItem is a good option to use.

Hope you like this article, if you have any questions, just drop a comment below.