How to Copy a File in PowerShell with Copy-Item

lazyadmin.nl/powershell/copy-file

April 26, 2022

We can use PowerShell to copy a file from one location to another with the Copy-Item cmdlet. The location can be another local folder or a remote computer. Besides files, we can also copy complete folders with subfolders and content. And it's even possible to filter the files that we copy.

In this article, we are going to take a look at different examples of how to copy a file with PowerShell.

Copy a file with PowerShell

Let's start with the basics when it comes to copying files with PowerShell. When you want to copy a single file you will need to specify the source path and the destination folder:

```
Copy-Item -Path "C:\temp\files\la-ams-ad01-log-1.txt" -Destination "d:\temp\"
# Copy is a shorthand for Copy-Item:
Copy "C:\temp\files\la-ams-ad01-log-1.txt" "d:\temp\"
```

Keep in mind that you will need to end the destination folder name with a \, otherwise, you will rename the file. This is also one of the useful features of the `Copy-Item` cmdlet, we can specify a new name for the file on the destination:

```
# Copy and rename the file:
Copy-Item -Path "C:\temp\files\la-ams-ad01-log-2.txt" -Destination "d:\temp\logfile.txt"
```

Copy Multiple Files

To copy multiple files with PowerShell, you can either use filters, which I will explain later in the article, or select multiple files in the path. To do this you will need to separate each file with a comma. For the destination, you will need to use a folder.

```
# Copy ad01-log-1.txt and ad01-log-2.txt
Copy-Item -Path C:\temp\files\ad01-log-1.txt,C:\temp\files\ad01-log-2.txt -Destination
d:\temp\
```

PowerShell Copy Folder

We can use the Copy-Item cmdlet also to copy a complete folder to a new location. But when it comes to copying folders it's important to use the correct ending in the source and destination paths:

Source	Destination	Results
c:\temp	d:\temp	Creates only the folder temp
c:\temp\	d:\temp\	Creates only the folder temp
c:\temp*	d:\temp	Create folder temp and copy only the files from c:\temp
c:\temp\	d:\temp -Recurse	Copies folder inc subfolders and all content

Copy Folders with PowerShell

In the examples below I am going to use the following folder structure with a subfolder as the source:

```
C:\TEMP\FILES
│ 11022022-app.log
│ 11032021-app.log
│ 11032022-app.log
│ 11102021-app.log
│ 15032022-app.log
│ la-ams-ad01-log-1.txt
│ la-ams-ad01-log-2.txt
│ la-ams-ad01-log-3.txt
│ la-ams-ad01-log-4.txt
│ la-ams-file02-log-1.txt
│ la-ams-file02-log-2.txt
│
└───subfolder
la-osl-ad01-log-1.txt
la-osl-file01-log-1.txt
LT3452-process-errors.log
LT3452-process.log
readme.txt
```

To copy only the files and empty subfolders from `c:\temp\files` we will need to use a wildcard for the source path:

```
Copy-Item -Path "C:\temp\files\*" -Destination "d:\temp\"
# Result
D:\TEMP
│ 11022022-app.log
│ 11032021-app.log
│ 11032022-app.log
│ 11102021-app.log
│ 15032022-app.log
│ la-ams-ad01-log-1.txt
```

```
│ la-ams-ad01-log-2.txt
│ la-ams-ad01-log-3.txt
│ la-ams-ad01-log-4.txt
│ la-ams-file02-log-1.txt
│ la-ams-file02-log-2.txt
│
└─────subfolder # Note - Sub folder is empty
```

To copy all content including the subfolders with PowerShell we will need to specify the `-recurce` parameter. This way the content of the subfolder is also copied to the new location:

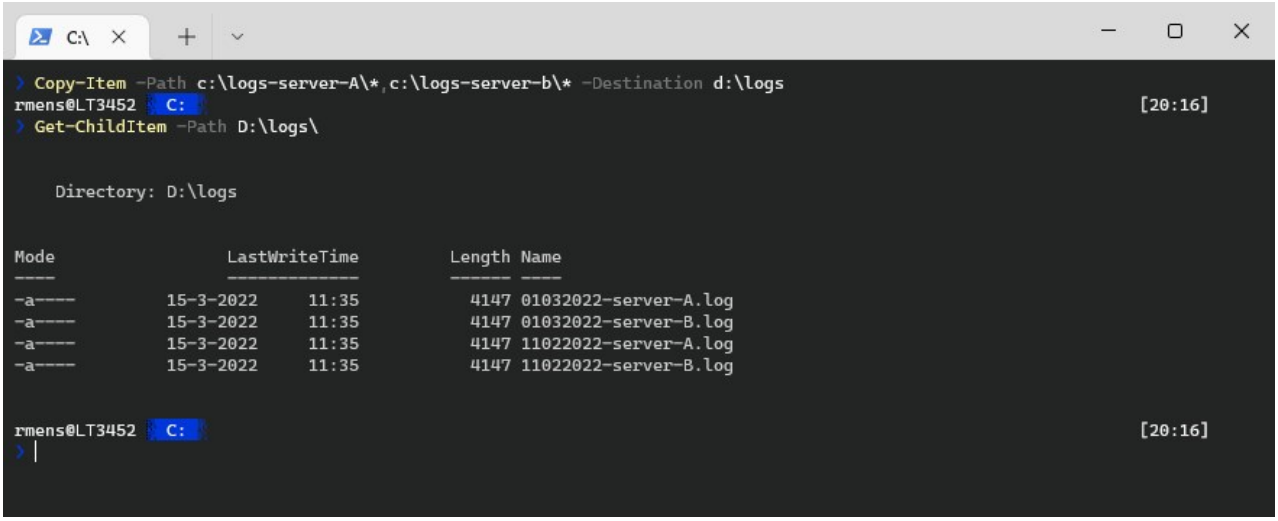
```
# Copy all content including subfolder:
Copy-Item -Path "C:\temp\files\*" -Destination "d:\temp\" -Recurse
# Result
D:\TEMP
│ 11022022-app.log
│ 11032021-app.log
│ 11032022-app.log
│ 11102021-app.log
│ 15032022-app.log
│ la-ams-ad01-log-1.txt
│ la-ams-ad01-log-2.txt
│ la-ams-ad01-log-3.txt
│ la-ams-ad01-log-4.txt
│ la-ams-file02-log-1.txt
│ la-ams-file02-log-2.txt
│
└─────subfolder
la-osl-ad01-log-1.txt
la-osl-file01-log-1.txt
LT3452-process-errors.log
LT3452-process.log
readme.txt
```

Copy and Merge Multiple Folders at Once

Something you probably won't use a lot but can be very handy is the ability to copy and merge multiple folders at once with the `Copy-Item` cmdlet. We can specify multiple source paths to the cmdlet and a single destination.

This way the contents of all source locations will be copied to a single location:

```
Copy-Item -Path c:\logs-server-A\*,c:\logs-server-b\* -Destination d:\logs
```



Filter Files to Copy

With the `Include` and `Exclude` parameters we can filter out which files to copy or not. The filter is matched against the full name of the file (including extension) and against folder names in your source path.

So to copy only files with the extension `.log` we can use the include filter like this:

```
# Copy only files with the extension .log
Copy-Item -Path c:\temp\* -Destination d:\temp -Include *.log
# Result
# Note that the .log files from the subfolder are note copied
D:\TEMP
11022022-app.log
11032021-app.log
11032022-app.log
11102021-app.log
15032022-app.log
```

Note the wildcard, which is always required if you want to filter on a part of the name. Good to know is that the `-Recurse` parameter won't work here. If you have subfolders with `.log` files in them, they won't be copied. The filter is only applied to the top level of the path.

Using the Exclude Filter

Excluding files works the same way. We can specify a part of the name of the files and folders that we don't want to copy. Keep in mind, that if a subfolder doesn't match the exclude string, it will be copied as well with all contents.

```
# Copy everything, except .log files:
Copy-Item -Path c:\temp\files\* -Destination d:\temp\ -Exclude *.log -Recurse
# Result:
D:\TEMP
│ la-ams-ad01-log-1.txt
```

```
│   la-ams-ad01-log-2.txt
│   la-ams-ad01-log-3.txt
│   la-ams-ad01-log-4.txt
│   la-ams-file02-log-1.txt
│   la-ams-file02-log-2.txt
│
└───subfolder
la-osl-ad01-log-1.txt
la-osl-file01-log-1.txt
readme.txt
```

Copy File to Remote Computer with PowerShell

It's also possible to copy files to and from a remote computer with PowerShell. There are two ways to do this, you could create a temporary network drive, using the New-PSDrive cmdlet. Or by first creating a new PowerShell session to a remote computer.

For this to work, you will need to have access to the remote computer, or use the credentials of the remote computer to start a PSSession:

```
$Session = New-PSSession -ComputerName "LazyServer01" -Credential "LazyAdmin\User01"
```

With the sessions created and stored in the `$session` variable, we can copy items to any location on the remote computer. The destination path, in this case, is the local path on the remote computer, so to copy files to the temp folder on the remote computer, you can simply use c:\temp as the destination path:

```
# Copy all files from the local temp folder to the temp folder on the remote computer
Copy-Item -Path c:\temp\local\* -Destination c:\temp\remote\ -ToSession $Session -Recurse
```

You can use the same method to copy files from a remote computer to your local computer with PowerShell. In this case, we only need to use the `-FromSession` parameter instead of the `-ToSession`.

```
$Session = New-PSSession -ComputerName "LazyServer01" -Credential "LazyAdmin\User01"
Copy-Item -Path c:\temp\remote* -Destination c:\temp\local\ -FromSession $Session -Recurse
```

Wrapping Up

The PowerShell Copy-Item cmdlet is really useful when it comes to copying files on your local or remote computer. You can't compare it with the powerful robocopy command, but nevertheless, it's still a useful cmdlet.

I hope this article helped you to copy a file or folder with PowerShell. If you have any questions, just drop a comment below.

Did you **Liked** this **Article**?
Get the latest articles like this **in your mailbox**
or share this article

I hate spam to, so you can unsubscribe at any time.