

SEARCH USING POWERSHELL

NAME SEARCH

Search file/folder NAMES using "-Filter"

```
PS C:\> ls -Path "C:\logs" -Filter "*local*" -Recurse -File | Select FullName
```

- Path: Either use "-Path 'C:\MyPath\2\Search' " OR just cd to the folder you want to search beforehand
- File: Searches for files only
- Directory: Searches for directories only
- Recurse: Searches subfolders too (without it, subfolders are excluded)
- Force: Includes hidden files/folders in search

- `ls`, `dir`, `gci`, are all just aliases for `Get-ChildItem`. See them for yourself in PowerShell using "`alias -Definition Get-ChildItem`"

CONTENTS SEARCH

Search file CONTENTS by piping `Get-ChildItem` results to `Select-String`

```
# Show occurrences of "DPI" within each file where it's found
PS C:\> ls -File -Recurse | Select-String "DPI"

# Show file(s) containing the string, but only show each file once (AKA summary)
PS C:\> ls -File | Select-String -Pattern "DPI" | Select-Object -Unique Path
```

- Pattern: It's optional to spell this out (i.e., "`Select-String -Pattern DPI`" is the same as "`Select-String DPI`").
- AllMatches: A `Select-String` option. The default action of `Select-String`, is to only find the first occurrence on each line of text. Using this however, finds all matches in each line (multiple occurrences per line). It may make no noticeable difference in PowerShell 5.1, but in PowerShell 7.x, you'll notice a difference in the highlighting.
- `select` is an alias for `Select-Object` and `sls` is an alias for `Select-String`.
- "`| Select-Object -Unique Path`" is handy when you just want to list the file(s) that contain the string, and not detail out where each occurrence is. Try it and you'll see what I mean.