PowerShell File and Directory Operations Tutorial

Table of Contents

Getting Started

PowerShell uses cmdlets (command-lets) that follow a Verb-Noun pattern. Most file operations use aliases that are familiar to users of other command-line interfaces.

Opening PowerShell

- **Windows 10/11**: Press `Win + X` and select "Windows PowerShell" or "Terminal"
- **Search**: Type "PowerShell" in the Start menu
- **Run dialog**: Press `Win + R`, type `powershell`, and press Enter

Navigation Commands

Get Current Location

```powershell
# Show current directory
Get-Location
# or use the alias
pwd
```

Change Directory

```powershell
# Change to a specific directory
Set-Location "C:\Users\YourName\Documents"
# or use the alias
cd "C:\Users\YourName\Documents"

# Go up one directory level
cd ..

# Go to the root of the current drive
cd \

# Change to home directory
cd ~
# or
cd $HOME

# Go back to previous directory
cd -
```

Useful Navigation Tips

- Use tab completion: Type part of a path and press Tab
- Use wildcards: `cd D*` to go to first directory starting with "D"
- Use relative paths: `cd .\subfolder` or `cd subfolder`

Viewing Directory Contents

List Files and Directories

powershell

```powershell
# List items in current directory
Get-ChildItem
# or use aliases
ls
dir

# List all items including hidden files
Get-ChildItem -Force
ls -Force

# List only directories
Get-ChildItem -Directory
ls -Directory

# List only files
Get-ChildItem -File
ls -File

# List with detailed information
Get-ChildItem | Format-List
ls | fl

# List files recursively (including subdirectories)
Get-ChildItem -Recurse
ls -Recurse
```

Filtering Listings

powershell

```powershell
# List files with specific extension
Get-ChildItem *.txt
ls *.txt

# List files matching a pattern
Get-ChildItem *report*
ls *report*

# List files in a specific directory
Get-ChildItem "C:\Windows\System32" -Filter "*.exe"
```

Creating Files and Directories

Create Directories

```powershell
# Create a single directory
New-Item -ItemType Directory -Name "MyFolder"
# or use alias
mkdir "MyFolder"

# Create nested directories
New-Item -ItemType Directory -Path "Projects\WebApp\Assets" -Force
mkdir "Projects\WebApp\Assets" -Force

# Create multiple directories
mkdir "Folder1", "Folder2", "Folder3"
```

Create Files

```powershell
# Create an empty file
New-Item -ItemType File -Name "example.txt"
# or
ni example.txt

# Create a file with content
New-Item -ItemType File -Name "readme.txt" -Value "This is a readme file"

# Create multiple files
New-Item -ItemType File -Name "file1.txt", "file2.txt", "file3.txt"
```

Copying, Moving, and Renaming

Copy Files and Directories

powershell

```powershell
# Copy a file
Copy-Item "source.txt" "destination.txt"
# or use alias
cp "source.txt" "destination.txt"

# Copy a file to another directory
Copy-Item "example.txt" "C:\Backup\"

# Copy a directory and all contents
Copy-Item "SourceFolder" "DestinationFolder" -Recurse

# Copy multiple files
Copy-Item "*.txt" "C:\TextFiles\"

# Copy with confirmation
Copy-Item "important.txt" "backup.txt" -Confirm
```

Move Files and Directories

powershell

```powershell
# Move a file
Move-Item "oldlocation.txt" "newlocation.txt"
# or use alias
mv "oldlocation.txt" "newlocation.txt"

# Move to another directory
Move-Item "myfile.txt" "C:\Documents\"

# Move multiple files
Move-Item "*.log" "C:\Logs\"

# Move a directory
Move-Item "OldFolder" "C:\NewLocation\"
```

Rename Files and Directories

```powershell
# Rename a file
Rename-Item "oldname.txt" "newname.txt"
# or use alias
ren "oldname.txt" "newname.txt"

# Rename a directory
Rename-Item "OldFolderName" "NewFolderName"

# Rename with wildcards (change extension)
Get-ChildItem *.txt | Rename-Item -NewName {$_.name -replace '\.txt$','.bak'}
```

Deleting Files and Directories

Remove Files

```powershell
# Delete a single file
Remove-Item "filename.txt"
# or use alias
rm "filename.txt"
del "filename.txt"

# Delete multiple files
Remove-Item "*.tmp"

# Delete with confirmation
Remove-Item "important.txt" -Confirm

# Force delete (including read-only files)
Remove-Item "protected.txt" -Force
```

Remove Directories

```powershell
# Delete an empty directory
Remove-Item "EmptyFolder"
rmdir "EmptyFolder"

# Delete directory and all contents
Remove-Item "FolderWithFiles" -Recurse

# Delete with confirmation
Remove-Item "ImportantFolder" -Recurse -Confirm

# Force delete everything
Remove-Item "StubborrnFolder" -Recurse -Force
```

File Content Operations

View File Contents

```powershell
# Display entire file content
Get-Content "filename.txt"
# or use alias
cat "filename.txt"
type "filename.txt"

# Display first few lines
Get-Content "filename.txt" -Head 10

# Display last few lines
Get-Content "filename.txt" -Tail 10

# Monitor file for changes (like tail -f)
Get-Content "logfile.txt" -Wait
```

Create and Modify File Content

```powershell
# Write content to file (overwrites existing)
"Hello World" | Set-Content "greeting.txt"
"Hello World" > "greeting.txt"

# Append content to file
"New line" | Add-Content "greeting.txt"
"New line" >> "greeting.txt"

# Write multiple lines
@"
Line 1
Line 2
Line 3
"@ | Set-Content "multiline.txt"
```

Search Within Files

```powershell
# Search for text in a file
Select-String -Pattern "error" -Path "logfile.txt"

# Search in multiple files
Select-String -Pattern "TODO" -Path "*.cs"

# Case-insensitive search
Select-String -Pattern "Error" -Path "*.log" -CaseSensitive:$false

# Search recursively in directories
Select-String -Pattern "password" -Path "C:\Scripts\*" -Recurse
```

File and Directory Properties

Get Item Information

```powershell
# Get detailed information about an item
Get-Item "filename.txt"
Get-Item "FolderName"

# Get file size, creation date, etc.
Get-ItemProperty "filename.txt"

# Get specific properties
(Get-Item "filename.txt").Length  # File size
(Get-Item "filename.txt").CreationTime
(Get-Item "filename.txt").LastWriteTime
```

Modify File Attributes

```powershell
# Set file as read-only
Set-ItemProperty "filename.txt" -Name IsReadOnly -Value $true

# Remove read-only attribute
Set-ItemProperty "filename.txt" -Name IsReadOnly -Value $false

# Set file attributes
Set-ItemProperty "filename.txt" -Name Attributes -Value "Hidden"
```

Searching and Filtering

Find Files and Directories

```powershell
# Find files by name
Get-ChildItem -Name "*.pdf" -Recurse

# Find files modified in last 7 days
Get-ChildItem | Where-Object {$_.LastWriteTime -gt (Get-Date).AddDays(-7)}

# Find large files (over 100MB)
Get-ChildItem -Recurse | Where-Object {$_.Length -gt 100MB}

# Find empty directories
Get-ChildItem -Directory -Recurse | Where-Object {(Get-ChildItem $_.FullName).Count -eq 0}

# Find files by extension and size
Get-ChildItem -Filter "*.log" | Where-Object {$_.Length -gt 1MB}
```

Advanced Filtering

```powershell
# Combine multiple conditions
Get-ChildItem | Where-Object {$_.Extension -eq ".txt" -and $_.Length -gt 1KB}

# Sort results
Get-ChildItem | Sort-Object Length -Descending

# Group results
Get-ChildItem | Group-Object Extension
```

Practice Exercises

Complete these exercises to master PowerShell file and directory operations. Create a practice directory first: `mkdir PowerShellPractice` and `cd PowerShellPractice`.

Exercise 1: Basic Navigation and Listing

1. Navigate to your home directory
2. List all files and folders, including hidden ones
3. Navigate to the Desktop
4. Go back to the previous directory
5. Show your current location

Solution:

```powershell
cd ~
ls -Force
cd Desktop
cd -
pwd
```

Exercise 2: Creating Directory Structure

Create the following directory structure:

```
Projects/
├── WebApp/
│   ├── css/
│   ├── js/
│   └── images/
├── MobileApp/
│   ├── android/
│   └── ios/
└── Documentation/
```

Solution:

```powershell
mkdir Projects
cd Projects
mkdir WebApp, MobileApp, Documentation
cd WebApp
mkdir css, js, images
cd ../MobileApp
mkdir android, ios
cd ..
```

Exercise 3: File Creation and Content Management

1. In the Documentation folder, create a file called "README.md"
2. Add the text "# Project Documentation" to the file
3. Create three more files: "todo.txt", "notes.txt", and "changelog.md"

4. Add some sample content to each file
5. List all files in the Documentation folder

Solution:

```powershell
cd Documentation
"# Project Documentation" > README.md
"- Task 1`n- Task 2" > todo.txt
"Meeting notes from today" > notes.txt
"## Version 1.0`nInitial release" > changelog.md
ls
```

Exercise 4: Copying and Moving Operations

1. Copy README.md to the WebApp folder
2. Copy all .txt files to a new folder called "TextFiles"
3. Move changelog.md to the Projects root directory
4. Create a backup copy of todo.txt called "todo_backup.txt"

Solution:

```powershell
cp README.md ../WebApp/
mkdir TextFiles
cp *.txt TextFiles/
mv changelog.md ../
cp todo.txt todo_backup.txt
```

Exercise 5: File Search and Filtering

1. Find all .txt files in the entire Projects directory structure
2. Find all files larger than 0 bytes
3. Find all directories that contain the word "app" (case-insensitive)
4. List all files sorted by modification time (newest first)

Solution:

```powershell
cd ../..  # Go back to Projects root
Get-ChildItem -Filter "*.txt" -Recurse
Get-ChildItem -Recurse | Where-Object {$_.Length -gt 0}
Get-ChildItem -Directory -Recurse | Where-Object {$_.Name -like "*app*"}
Get-ChildItem -File -Recurse | Sort-Object LastWriteTime -Descending
```

Exercise 6: Content Search and Manipulation

1. Search for the word "Project" in all files
2. Add a new line "Updated today" to the end of README.md
3. Display the first 3 lines of README.md
4. Count the number of lines in todo.txt

Solution:

```powershell
Select-String -Pattern "Project" -Path * -Recurse
"Updated today" >> Documentation/README.md
Get-Content Documentation/README.md -Head 3
(Get-Content Documentation/todo.txt | Measure-Object -Line).Lines
```

Exercise 7: Cleanup Operations

1. Delete all .txt files in the TextFiles folder
2. Remove the empty TextFiles folder
3. Create a compressed backup of the WebApp folder (if you know how)
4. Rename the MobileApp folder to "MobileApps"

Solution:

```powershell
Remove-Item Documentation/TextFiles/*.txt
Remove-Item Documentation/TextFiles
# Compression requires additional cmdlets - this is advanced
Rename-Item MobileApp MobileApps
```

Exercise 8: Advanced Challenge

Create a script that:

1. Creates 10 files named "file01.txt" through "file10.txt"
2. Adds the current date and time to each file
3. Finds files with numbers 05-08 in their names
4. Copies these files to a new "Selected" folder
5. Lists the contents of the Selected folder with file sizes

Solution:

```powershell
# Create 10 files with current date/time
1..10 | ForEach-Object {
    $filename = "file{0:00}.txt" -f $_
    Get-Date | Set-Content $filename
}

# Find files 05-08 and copy to Selected folder
mkdir Selected
Get-ChildItem file0[5-8].txt | Copy-Item -Destination Selected/

# List contents with sizes
Get-ChildItem Selected/ | Select-Object Name, Length
```

Tips for Success

1. **Use Tab Completion**: Press Tab to auto-complete file and directory names
2. **Use Aliases**: Learn common aliases like `ls`, `cd`, `cp`, `mv`, `rm`
3. **Read Help**: Use `Get-Help` followed by any cmdlet name for detailed information
4. **Practice Regularly**: These operations become second nature with practice
5. **Use ISE or VS Code**: Consider using PowerShell ISE or VS Code for longer scripts
6. **Backup Important Data**: Always backup before performing bulk operations

Next Steps

After mastering these basics, explore:

- PowerShell scripting and functions
- Working with CSV and JSON files

- Remote file operations
- PowerShell modules for extended functionality
- Advanced filtering with regular expressions

Remember: PowerShell is case-insensitive for commands and file paths, but it's good practice to maintain consistent casing for readability.