

PYTHON Schattenkopie- Management

Ich habe eine ziemlich große Festplatte und deshalb auch die File History von Windows ziemlich aufgebohrt. Zusammen mit der [Bilder-Stapelverarbeitung](#) kamen dann allerdings deutlich größere Berge an veränderten Dateien zutage, so dass die File History dann doch noch in Richtung Flut ging.

Jetzt weiß ich natürlich nicht warum Windows das Cleanup nicht machen will nach Umstellung der Aufbewahrungsrichtlinien.

Muss ich auch nicht wissen... Ist irgendwie so...

```
import os
import shutil
from datetime import datetime, timedelta
import concurrent.futures

def get_drives():
    """Listet verfügbare Laufwerke auf."""
    import psutil
    drives = [partition.mountpoint for partition in psutil.disk_partitions() if 'fixed' in
partition.opts.lower()]
    return drives

def select_drive(drives):
    """Benutzer wählt Laufwerk aus."""
    print("Verfügbare Laufwerke:")
    for i, drive in enumerate(drives, 1):
        total, used, free = shutil.disk_usage(drive)
        print(f"{i}. {drive} - Größe: {total // (1024**3)} GB")
    choice = int(input("Wählen Sie das Laufwerk (Nummer): ")) - 1
    return drives[choice]

def is_old_file(file_path, threshold_days=90):
    """Prüft, ob Datei älter als Schwellenwert ist."""
    file_mtime = datetime.fromtimestamp(os.path.getmtime(file_path))
```

```

        return datetime.now() - file_mtime > timedelta(days=threshold_days)

def process_files(directory, threshold_days=90):
    """Durchsucht Dateien parallel."""
    old_files = []
    total_size = 0

    with concurrent.futures.ThreadPoolExecutor() as executor:
        file_futures = []
        for root, _, files in os.walk(directory):
            for file in files:
                file_path = os.path.join(root, file)
                file_futures.append(executor.submit(is_old_file, file_path, threshold_days))

        for future, file_path in zip(concurrent.futures.as_completed(file_futures),
                                     [os.path.join(root, file) for root, _, files in
                                     os.walk(directory) for file in files]):
            if future.result():
                try:
                    file_size = os.path.getsize(file_path)
                    old_files.append((file_path, file_size))
                    total_size += file_size
                except Exception as e:
                    print(f"Fehler bei {file_path}: {e}")

    return old_files, total_size

def main():
    drives = get_drives()
    selected_drive = select_drive(drives)
    directory = os.path.join(selected_drive, 'FileHistory')

    print("Suche nach alten Dateien...")
    old_files, total_size = process_files(directory)

    print(f"\nGefundene alte Dateien: {len(old_files)}")
    print(f"Gesamtgröße: {total_size / (1024**3):.2f} GB")

    for file, size in old_files[:10]: # Zeige erste 10 Dateien
        print(f"{file} - {size / (1024**2):.2f} MB")

```

```

confirm = input("\nMöchten Sie diese Dateien löschen? (j/n): ")
if confirm.lower() == 'j':
    for file, _ in old_files:
        try:
            os.remove(file)
        except Exception as e:
            print(f"Fehler beim Löschen von {file}: {e}")
    print("Dateien gelöscht.")

if __name__ == "__main__":
    main()

```

Festplatte auswählen, dann wird ./FileHistory durchsucht und alles erst mal ausgegeben was älter als 90 Tage ist. Dann gibts ne Option zum Löschen.

Hier müsst ihr noch das python Modul psutil installieren.

Das ganze geht auch in Powershell (dann halt nur langsamer)

```

# Skript zur Datei-Bereinigung mit Fortschrittsanzeige und Festplattenauswahl
Clear-Host

# Alle Festplatten auflisten
$drives = Get-PSDrive -PSProvider FileSystem |
    Where-Object {$_.Free -and $_.Used} |
    Select-Object Root, @{Name='SizeGB';Expression={[math]::Round(($_ .Used + $_ .Free)/1GB,
2)}}

# Festplattenauswahl
Write-Host "Verfügbare Festplatten:"
for ($i=0; $i -lt $drives.Count; $i++) {
    Write-Host "$($i+1). $($drives[$i].Root) - $($drives[$i].SizeGB) GB"
}

$driveChoice = Read-Host "Wählen Sie die Festplatte (Nummer)"
$selectedDrive = $drives[$driveChoice-1].Root

# Parameter
$path = "$selectedDrive\FileHistory"

```

```

$thresholdDays = 90

# Fortschrittsanzeige
function Show-Progress {
    param($total, $current, $activity)
    $percentage = [math]::Floor(($current / $total) * 100)
    Write-Progress -Activity $activity -Status "$percentage% Complete" -PercentComplete
    $percentage
}

# Dateien sammeln
$allFiles = Get-ChildItem -Path $path -Recurse -File
$total = $allFiles.Count
$oldFiles = @()

# Durchsuchen mit Fortschrittsanzeige
for ($i = 0; $i -lt $total; $i++) {
    $file = $allFiles[$i]
    Show-Progress -total $total -current $i -activity "Dateien durchsuchen"

    if ($file.LastWriteTime -lt (Get-Date).AddDays(-$thresholdDays)) {
        $oldFiles += $file
    }
}

# Ergebnisse anzeigen
Write-Host "`nGefundene alte Dateien:"
$oldFiles | Format-Table FullName, LastWriteTime,
@{Name='SizeMB';Expression={[math]::Round($_.Length/1MB,2)}}

# Bestätigung und Löschung
$confirm = Read-Host "Möchten Sie diese $($oldFiles.Count) Dateien wirklich löschen?
(j/n)"
if ($confirm -eq "j") {
    for ($i = 0; $i -lt $oldFiles.Count; $i++) {
        Show-Progress -total $oldFiles.Count -current $i -activity "Dateien löschen"
        Remove-Item $oldFiles[$i].FullName -Force
    }
    Write-Host "`nAlle alten Dateien wurden gelöscht."
}

```

```
} else {  
    Write-Host "Löschvorgang abgebrochen."  
}
```

Version #4

Erstellt: 2025-01-25 00:05:54 UTC von Konstantin

Zuletzt aktualisiert: 2025-06-05 14:44:46 UTC von Konstantin