Entwicklerdokumentation

Von Marek von Rogall  
Modul 122  
Abläufe mit einer Scriptsprache automatisieren

Allgemeines Seite 1

Anforderungen Seite 2

Der Algorithmus (als PAP) Seite 3

Der Code inkl. Kommentare und Header Seite 4-7

Testfälle Seite 8-12

Testprotokoll Seite 13-14

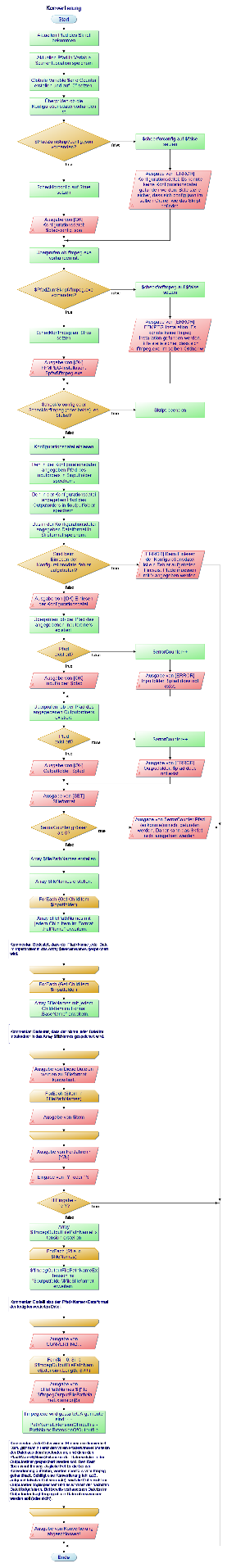
**Allgemeines:**

Verwendete PowerShell Version: 5.1.19041.3031  
Betriebssystem: Win10 Pro V 22H (B 19045.3086)  
Autor Dokumentationen & Skript: Marek von Rogall  
Endgültige Fertigstellung aller Dateien: 01.07.2023  
Fremdhilfe: https://stackoverflow.com/

**Anforderungen**

|  |  |  |
| --- | --- | --- |
| Anf.-Nr. | Typ | Anforderung |
| F1 | Funktional | Das Skript konvertiert Audio, Bild und Videodateien. |
| F2 | Funktional | Eingabe- & Ausgabeordner, sowie gewünschtes Dateiformat kann der Benutzer in einer Konfigurationsdatei frei wählen. |
| F3 | Funktional | Alle Dateien im Eingabeordner werden dem Benutzer vor dem Konvertieren aufgelistet und der Benutzer wird gefragt, ob er wirklich fortfahren möchte. |
| F4 | Funktional | Trigger: Das Skript wird manuell durch den Benutzer gestartet. (Rechtsklick > mit PowerShell ausführen / direkt über Konsole ausführen) |
| Q1 | Qualität | Eingabefehler sollen abgefangen werden. |
| Q2 | Qualität | Das Skript ist mit Kommentaren und einem Header versehen. |
| Q3 | Qualität | Es existiert eine Entwicklerdokumentation und eine Betriebsdokumentation. |
| R1 | Randbedingung | Das Skript muss innerhalb der vereinbarten Zeit fertiggestellt sein (inkl. Tests). |
| R2 | Randbedingung | Das Skript muss in PowerShell (mind. Version 5.1) geschrieben sein. |
| R3 | Randbedingung | Fremder Code und Fremdhilfe sind deklariert. |

**Der Algorithmus (als PAP)**



*(Das PAP habe ich Ihnen unter „vonRogallMarek\_PAP-Konvertierung.png“ im Ordner „vonRogallMarek\_LB\_M122\_2022-V3.zip“ abgelegt, damit Sie es sich besser anschauen können.)*

**Der Code inklusive Kommentare und Header:**

<# Skriptname: LA\_122\_vonRogall\_Konvertierer.ps1

Autor: Marek von Rogall

Date: 2023-07-01

Version: 1.0

Description: Konvertiert Audio, Bild und Videodateien in ein gewünschtes Dateiformat.

#>

Clear-Host

$currentLocation = $PSScriptRoot

$global:errorCounter = 0

<#Funktion, welche überprüft ob ein Pfad existiert (und das Ergebnis in der Konsole ausgibt)#>

*Function* CheckPath($pathToCheck, [*string*]$name) {

    if ((Test-Path $pathToCheck) -eq $true) {

        Write-Host -NoNewLine "[OK] " -ForegroundColor green

        Write-Host -NoNewLine $name -ForegroundColor cyan

        Write-Host $pathToCheck -ForegroundColor white

    }

    else {

        $global:errorCounter++

        Write-Host -NoNewLine "[ERROR] " -ForegroundColor red

        Write-Host -NoNewLine $name -ForegroundColor cyan

        Write-Host -NoNewLine $pathToCheck -ForegroundColor white

        Write-Host " does not exist" -ForegroundColor cyan

    }

}

<#Überprüfen, ob sich die Konfigurationsdatei im Ordner des Skripts befindet.#>

$checkforconfig = Test-Path "$currentLocation\config.json"

if ($checkforconfig -eq $false) {

    Write-Host -NoNewLine "[ERROR] " -ForegroundColor red

    Write-Host -NoNewLine "Konfigurationsdatei: " -ForegroundColor cyan

    Write-Host "Es konnte keine Konfigurationsdatei gefunden werden. Bitte stelle sicher, dass sich config.json im selben Ordner wie das Skript befindet." -ForegroundColor white

}

else {

    Write-Host -NoNewLine "[OK] " -ForegroundColor green

    Write-Host -NoNewLine "Konfigurationsdatei: " -ForegroundColor cyan

    Write-Host "$currentLocation\config.json" -ForegroundColor white

}

<#Überprüfen, ob sich ffmpeg.exe im Ordner des Skripts befindet.#>

$checkforffmpeg = Test-Path "$currentLocation\ffmpeg.exe"

if ($checkforffmpeg -eq $false) {

    Write-Host -NoNewLine "[ERROR] " -ForegroundColor red

    Write-Host -NoNewLine "FFMPEG-Installation: " -ForegroundColor cyan

    Write-Host "Es konnte keine ffmpeg Installation gefunden werden. Bitte stelle sicher, dass sich ffmpeg.exe im selben Ordner wie das Skript befindet." -ForegroundColor white

}

else {

    Write-Host -NoNewLine "[OK] " -ForegroundColor green

    Write-Host -NoNewLine "FFMPEG-Installation: " -ForegroundColor cyan

    Write-Host "$currentLocation\ffmpeg.exe" -ForegroundColor white

}

<#Konfigurationsdatei und/oder ffmpeg.exe nicht vorhanden = Skript beenden, ansonsten weiter ausführen.#>

if ($checkforconfig -eq $false -or $checkforffmpeg -eq $false) {

    Write-Host "`nAufgrund diverser Fehler kann das Skript nicht ausgefuehrt werden." -ForegroundColor red

    Pause

    exit

}

<#Probieren Input- & Outputordner, sowie gewünschtes Dateiformat aus der Konfigurationsdatei auszulesen. Bei Fehlern wird das Skript beendet.#>

try {

    $jsonFilePath = "$currentLocation\config.json"

    $jsonContent = Get-Content -Raw -Path $jsonFilePath  | ConvertFrom-Json

    $inputfolder = $jsonContent.inputfolder

    $outputfolder = $jsonContent.outputfolder

    $fileformat = $jsonContent.fileformat

    Write-Host -NoNewLine "[OK] " -ForegroundColor green

    Write-Host "Einlesen der Konfigurationsdatei." -ForegroundColor yellow

}

catch {

    Write-Host -NoNewLine "[ERROR] " -ForegroundColor red

    Write-Host "Beim Einlesen der Konfigurationsdatei ist ein Fehler aufgetreten. Hinweis: Pfade muessen mit `"\\`" angegeben werden." -ForegroundColor yellow

    Pause

    exit

}

<#Überprüfen ob die eingelesenen Pfade wirklich existieren.#>

CheckPath "$inputfolder" "Inputfolder: "

CheckPath "$outputfolder" "Outputfolder: "

Write-Host -NoNewLine "[SET]" -ForegroundColor Yellow

Write-Host $fileformat -ForegroundColor white

<#Existiert mindestens eines der Pfade nicht, so wird das Skript beendet.#>

if ($errorCounter -gt 0) {

    Write-Host "`n$errorCounter Pfad(e) konnte(n) nicht gefunden werden. Daher kann das Skript nicht ausgefuert werden." -ForegroundColor red

    Pause

    exit

}

<#Dateipfad (inkl. Name) & Einzelnen Dateinamen von allen Dateien im Inputordner herausfinden und in ein Array speichern.#>

$filePathNames = @()

$fileNames = @()

$filePathNames += Get-ChildItem $inputfolder | % { $\_.FullName }

$fileNames += Get-ChildItem $inputfolder | % { $\_.BaseName }

<#Meldung welche Dateien konvertiert werden sollen.#>

Write-Host "$("`nDiese Dateien werden zu ")$fileformat$(" konvertiert:")" -ForegroundColor Red

ForEach ( $item in $filePathNames) {

    Write-Host $item

}

<#Fragen, ob diese wirklich konvertiert werden sollen.#>

Write-Host "Fortfahren? [Y/N]" -ForegroundColor Red

$answer = Read-Host

if ($answer -ne "Y") { exit }

Clear-Host

<#Generiere Output: $OutputOrdner\$Dateiname\$Dateiformat --> Z.B. "C:\users\Marek\Videos\video.mov"#>

$ffmpegOutputFilePathNameExtension = @()

ForEach ($file in $fileNames) {

    $ffmpegOutputFilePathNameExtension += "$outputfolder\$file$fileformat"

}

Write-Host "CONVERTING..." -ForegroundColor Red

<#Konvertiere InputFile1 zu OutputFile1, dann InputFile2 zu OutputFile2 usw...#>

For ($i = 0; $i -lt $ffmpegOutputFilePathNameExtension.Length; $i++) {

    Write-Host $filePathNames[$i]" to "$ffmpegOutputFilePathNameExtension[$i] -ForegroundColor Yellow

    & "$currentLocation\ffmpeg.exe" -i $filePathNames[$i] $ffmpegOutputFilePathNameExtension[$i]

}

Clear-Host

<#Meldung, dass die Konvertierung abgeschlossen wurde#>

Write-Host "Konvertierung abgeschlossen!" -ForegroundColor Cyan

Pause

exit

## Testfallspezifikation

**Testfallnummer:** 1.1

**Anforderungsnummer:** F1

**Voraussetzungen:** In der Konfigurationsdatei wurde angegeben, Dateien zu „.mp3“ zu konvertieren. Im Inputorndner liegt die Datei „datei.mp4“ Das Skript wurde gestartet. Die Konfigurationsdatei und ffmpeg.exe wurden erfolgreich lokalisiert. Ein- & Ausgabeordner sowie das gewünschte Dateiformat wurden in der Konfigurationsdatei deklariert und vom Skript erfolgreich eingelesen und nach ihrer Gültigkeit überprüft. Dem Benutzer wurden die Dateien im Inputordner aufgelistet und der Benutzer hat die bestätigt, mit der Konvertierung fortzufahren.

**Eingabe:**

Keine Eingabe.

**Ausgabe:**

Das Skript konvertiert «datei.mp4» aus dem Inputordner zu «datei.mp3» und speichert diese Datei im Outputordner ab.

**Testfallnummer:** 1.2

**Anforderungsnummer:** F1

**Voraussetzungen:** In der Konfigurationsdatei wurde angegeben, Dateien zu „.mp3“ zu konvertieren. Im Inputorndner liegt die Datei „datei.txt“ Das Skript wurde gestartet. Die Konfigurationsdatei und ffmpeg.exe wurden erfolgreich lokalisiert. Ein- & Ausgabeordner sowie das gewünschte Dateiformat wurden in der Konfigurationsdatei deklariert und vom Skript erfolgreich eingelesen und nach ihrer Gültigkeit überprüft. Dem Benutzer wurden die Dateien im Inputordner aufgelistet und der Benutzer hat die bestätigt, mit der Konvertierung fortzufahren.

**Eingabe:**

Keine Eingabe.

**Ausgabe:**

Das Skript probiert «datei.txt» aus dem Inputordner zu «datei.mp3» zu konvertieren und diese im Outputordner abzuspeichern. Da ffmpeg aber keine Textdateien konvertieren kann, wird diese Datei übersprungen und die Konvertierung wird mit der nächsten Datei fortgesetzt (falls vorhanden).

**Testfallnummer:** 2.1

**Anforderungsnummer:** F2

**Voraussetzungen:** Das Skript wurde noch nicht gestartet. Der Benutzer öffnet die Konfigurationsdatei und gibt Eingabe- & Ausgabeordner, sowie das gewünschte Dateiformat an. Das Skript wurde gestartet.

**Eingabe:**

{

"inputfolder": "C:\\Users\\Marek von Rogall\\Documents\\Skript\\Video Input",

"outputfolder": "C:\\Users\\Marek von Rogall\\Documents\\Skript\\Video Output",

"fileformat": ".gif"

}

**Ausgabe:**

Keine Ausgabe.

**Testfallnummer:** 2.2

**Anforderungsnummer:** F2

**Voraussetzungen:** Das Skript wurde noch nicht gestartet. Der Benutzer öffnet die Konfigurationsdatei und gibt Eingabe- & Ausgabeordner, sowie das gewünschte Dateiformat an.

**Eingabe:**

{

"inputfolder": "C:\\dieserPfad\\existiertNicht",

"outputfolder": "C:\\Users\\Marek von Rogall\\Documents\\Skript\\Video Output",

"fileformat": ".gif"

}

**Ausgabe:**

Keine Ausgabe.

**Testfallnummer:** 2.3

**Anforderungsnummer:** F2

**Voraussetzungen:** Die gewünschten Einstellungen in der Konfigurationsdatei wurden vorgenommen. Konfiguration der Konfigurationsdatei siehe Testfall 2.1. Das Skript wurde gestartet. Die Konfigurationsdatei und ffmpeg.exe wurden erfolgreich lokalisiert. Die Konfigurationsdatei wird nun eingelesen und auf Fehler überprüft.

**Eingabe:**

Keine Eingabe.

**Ausgabe:**

[OK] Inputfolder: C:\Users\Marek von Rogall\Documents\Skript\Video Input

[OK] Outputfolder: C:\Users\Marek von Rogall\Documents\Skript\Video Output

[SET].gif

Das Skript kann fortgesetzt werden, da keine Fehler aufgetreten sind.

**Testfallnummer:** 2.4

**Anforderungsnummer:** F2

**Voraussetzungen:** Die gewünschten Einstellungen in der Konfigurationsdatei wurden vorgenommen. Konfiguration der Konfigurationsdatei siehe Testfall 2.2. Das Skript wurde gestartet. Die Konfigurationsdatei und ffmpeg.exe wurden erfolgreich lokalisiert. Die Konfigurationsdatei wird nun eingelesen und auf Fehler überprüft.

**Eingabe:**

Keine Eingabe.

**Ausgabe:**

[ERROR] Inputfolder: C:\dieserPfad\existiertNicht does not exist

[OK] Outputfolder: C:\Users\Marek von Rogall\Documents\Skript\Video Output

[SET].gif

Das Skript wird beendet, da ein Fehler aufgetreten ist.

**Testfallnummer:** 3.1

**Anforderungsnummer:** F3

**Voraussetzungen:** Das Skript wurde gestartet. Die Konfigurationsdatei und ffmpeg.exe wurden erfolgreich lokalisiert. Ein- & Ausgabeordner sowie das gewünschte Dateiformat wurden in der Konfigurationsdatei deklariert und vom Skript erfolgreich eingelesen und nach ihrer Gültigkeit überprüft. Dem Benutzer wurden die Dateien im Inputordner aufgelistet und wird nun aufgefordert die Konvertierung mit „Y“ zu bestätigen oder mit „N“ abzubrechen.

**Eingabe:**

„Y“

**Ausgabe:**

Das Skript wird fortgesetzt. Die Dateien werden konvertiert und in den Outputordner gespeichert.

**Testfallnummer:** 3.2

**Anforderungsnummer:** F3

**Voraussetzungen:** Das Skript wurde gestartet. Die Konfigurationsdatei und ffmpeg.exe wurden erfolgreich lokalisiert. Ein- & Ausgabeordner sowie das gewünschte Dateiformat wurden in der Konfigurationsdatei deklariert und vom Skript erfolgreich eingelesen und nach ihrer Gültigkeit überprüft. Dem Benutzer wurden die Dateien im Inputordner aufgelistet und wird nun aufgefordert die Konvertierung mit „Y“ zu bestätigen oder mit „N“ abzubrechen.

**Eingabe:**

„N“

**Ausgabe:**

Das Skript wird beendet. Es werden keine Dateien konvertiert.

**Testfallnummer:** 3.3

**Anforderungsnummer:** F3

**Voraussetzungen:** Das Skript wurde gestartet. Die Konfigurationsdatei und ffmpeg.exe wurden erfolgreich lokalisiert. Ein- & Ausgabeordner sowie das gewünschte Dateiformat wurden in der Konfigurationsdatei deklariert und vom Skript erfolgreich eingelesen und nach ihrer Gültigkeit überprüft. Dem Benutzer wurden die Dateien im Inputordner aufgelistet und wird nun aufgefordert die Konvertierung mit „Y“ zu bestätigen oder mit „N“ abzubrechen.

**Eingabe:**

„asdjaslidjasild23213“

**Ausgabe:**

Das Skript wird beendet. Es werden keine Dateien konvertiert.

**Testfallnummer:** 4.1

**Anforderungsnummer:** F4

**Voraussetzungen:** Das Skript wurde nicht gestartet. Der Benutzer hat das Skript vor sich im Datei-Explorer liegen.

**Eingabe:**

Rechts-klick auf das Skript > mit PowerShell ausführen.

**Ausgabe:**

Das Skript wird ausgeführt.

**Testfallnummer:** 5.1

**Anforderungsnummer:** Q1

**Voraussetzungen:** Das Skript ist gestartet, eine Eingabe wurde getätigt (Einlesen der Konfigurationsdatei). Die Konfigurationsdatei ist unvollständig und enthält keinen gültigen Pfad eines Inputordners (Mögliche Konfiguration der Konfigurationsdatei nach Testfall NR. 2.2).

**Eingabe:**

Einlesen der Konfigurationsdatei.

**Ausgabe:**

Das Skript wird aufgrund des Fehlerhaften Inhalts der Konfigurationsdatei beendet.

**Testfallnummer:** 6.1

**Anforderungsnummer:** Q2

**Voraussetzungen:** Das Skript ist mit Kommentaren und einem Header versehen.

**Eingabe:**

Keine Eingabe.

**Ausgabe:**

Keine Ausgabe.

**Testfallnummer:** 7.1

**Anforderungsnummer:** Q3

**Voraussetzungen:** Es existiert eine Entwicklerdokumentation und eine Betriebsdokumentation.

**Eingabe:**

Keine Eingabe.

**Ausgabe:**

Keine Ausgabe.

**Testfallnummer:** 8.1

**Anforderungsnummer:** R1

**Voraussetzungen:** Das Skript muss innerhalb der vereinbarten Zeit fertiggestellt sein (inkl. Tests).

**Eingabe:**

Keine Eingabe.

**Ausgabe:**

Keine Ausgabe.

**Testfallnummer:** 9.1

**Anforderungsnummer:** R2

**Voraussetzungen:** Das Skript muss in PowerShell (mind. Version 5.1) geschrieben sein.

**Eingabe:**

Keine Eingabe.

**Ausgabe:**

Keine Ausgabe.

**Testfallnummer:** 10.1

**Anforderungsnummer:** R3

**Voraussetzungen:** Fremder Code und Fremdhilfe sind deklariert.

**Eingabe:**

Keine Eingabe.

**Ausgabe:**

Keine Ausgabe.

### **Testumgebung**

26.06.2023 / Baden  
01.07.2023 / Bellikon  
02.07.2023 / Bellikon

Betriebssystem: Windows 10 Pro  
Version Betriebssystem: Version 22H (Build 19045.3086)  
PowerShell-Version: 5.1.19041.3031

### **Testprotokoll**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test-Nr. | Testfall-Nr. | Datum | Tester | Resultat | Bemerkung | Unterschrift |
| 1 | 1.1 | 26.06.23 | Lorenzo Lai | Funktioniert (Ausgabe wie erwartet) | / | Lorenzo Lai |
| 2 | 1.2 | 26.06.23 | Lorenzo Lai | Funktioniert (Ausgabe wie erwartet) | / | Lorenzo Lai |
| 3 | 2.1 | 26.06.23 | Cyril Lutziger | Funktioniert (Ausgabe wie erwartet) | / | Cyril Lutziger |
| 4 | 2.2 | 26.06.23 | Cyril Lutziger | Funktioniert (Ausgabe wie erwartet) | / | Cyril Lutziger |
| 5 | 2.3 | 26.06.23 | Cyril Lutziger | Funktioniert (Ausgabe wie erwartet) | / | Cyril Lutziger |
| 6 | 2.4 | 26.06.23 | Cyril Lutziger | Funktioniert (Ausgabe wie erwartet) | / | Cyril Lutziger |
| 7 | 3.1 | 26.06.23 | Lorenzo Lai | Funktioniert (Ausgabe wie erwartet) | / | Lorenzo Lai |
| 8 | 3.2 | 26.06.23 | Lorenzo Lai | Funktioniert (Ausgabe wie erwartet) | / | Lorenzo Lai |
| 9 | 3.3 | 26.06.23 | Lorenzo Lai | Funktioniert (Ausgabe wie erwartet) | / | Lorenzo Lai |
| 10 | 4.1 | 01.07.23 | Marek von Rogall | Funktioniert (Ausgabe wie erwartet) | Trigger funktioniert | Marek von Rogall |
| 11 | 5.1 | 01.07.23 | Marek von Rogall | Funktioniert (Ausgabe wie erwartet) | Fehler werden abgefangen. | Marek von Rogall |
| 12 | 6.1 | 01.07.23 | Marek von Rogall | Funktioniert (Ausgabe wie erwartet) | Kommentare und Header vorhanden | Marek von Rogall |
| 13 | 7.1 | 01.07.23 | Marek von Rogall | Funktioniert (Ausgabe wie erwartet) | Dokumentationen vorhanden. | Marek von Rogall |
| 14 | 8.1 | 01.07.23 | Marek von Rogall | Funktioniert (Ausgabe wie erwartet) | Das Skript und alle Dokumentationen wurden fertiggestellt | Marek von Rogall |
| 15 | 9.1 | 26.06.23 | Marek von Rogall | Funktioniert (Ausgabe wie erwartet) | Das Skript wurde in mind. PowerShell 5.1 geschrieben. | Marek von Rogall |
| 16 | 10.1 | 01.07.23 | Marek von Rogall | Funktioniert (Ausgabe wie erwartet) | Fremdhilfe wurde deklariert. (Seite 1) | Marek von Rogall |

Eingeholtes Feedback eines Lernpartners: