



Anwendungsarbeit

POWERSHELL BACKUP

Bernhard Tim, Wang Hao

29.Nov.2021



AUFTRAG	2
PAP	3
TESTFÄLLE	4
1. TESTFALL	4
2. TESTFALL	4
3. TESTFALL	4
BEDIENUNGSANLEITUNG	5
1. SCHRITT	5
2. SCHRITT	5
3. SCHRITT	5
4. SCHRITT	5
5. SCHRITT	5
FAZIT	6
ANHÄNGE	7
POWERSHELL SKRIPT	7
LOG-DATEI	9
DOKUMENTATIONSKONZEPT	10
A) SKRIPTFILES	10
B) FUNKTIONEN	10

Auftrag

Unser Auftrag war eine einfache Backuplösung einen definierten Anzahl Verzeichnisse und Dateien in PowerShell zu implementieren. Unter den Vorgaben, dass das Backup eine Kopie des entsprechenden Verzeichnisses oder Datei sein soll. Bei der Erstellung des Backups sollte eine Log Datei erstellt und gespeichert werden. Die Backups sollten überprüfbar sein und es dürfen keine «Drittkomponenten» gebraucht werden wie bspw. Backup-Tools. Alles sollte in einem AA-Bericht abgegeben werden was das PowerShell Skript, PAP, Log und die Testfälle enthält. Dazu sollte man noch ein Video aufnehmen was erklärt wie wir vorgegangen sind, wie das Programm funktioniert und die Testfälle zeigt. Detaillierter Auftrag sehen sie im Bild unten.

Auftrag

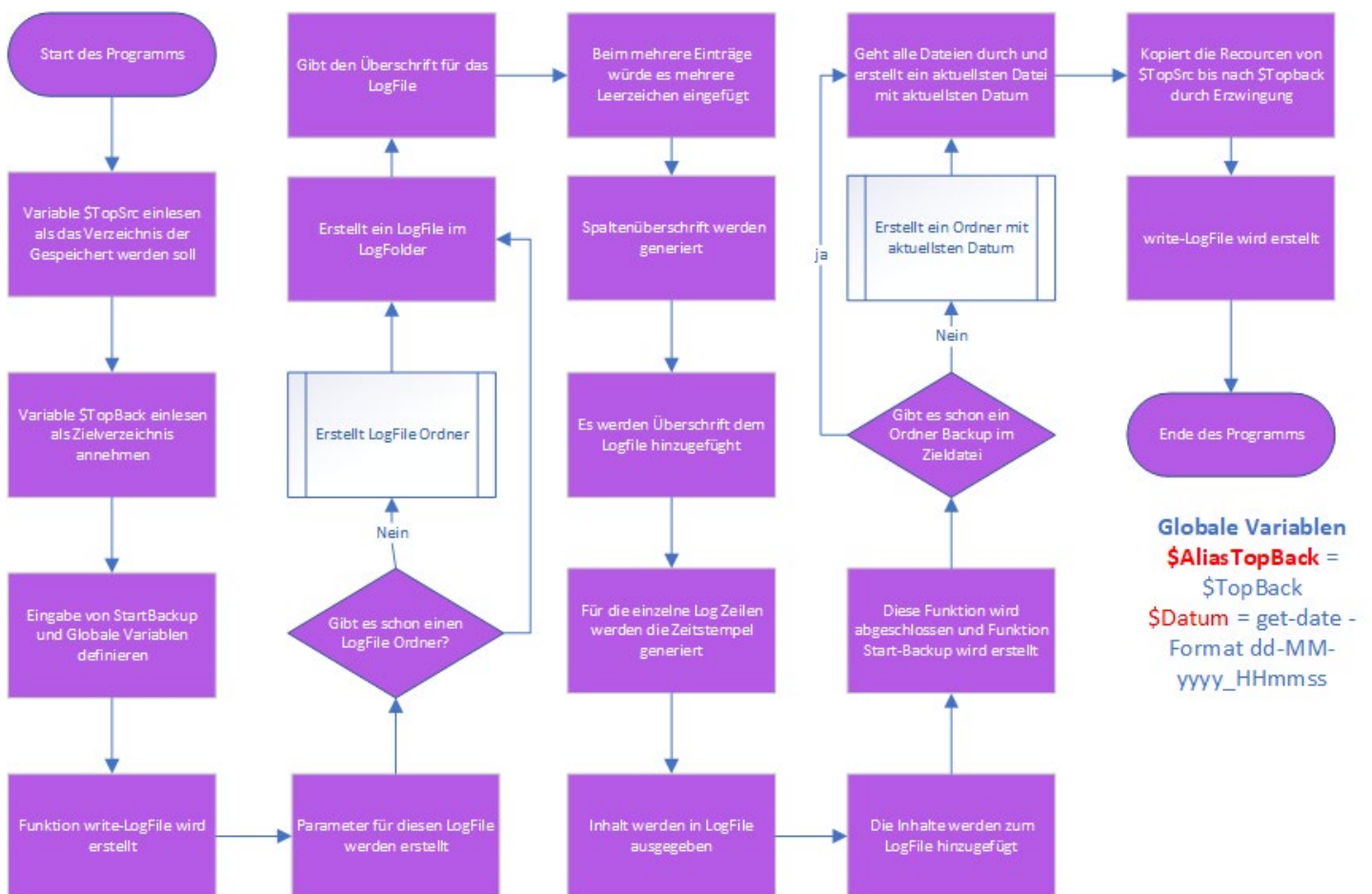
- 1) Sie sollen eine **einfache Backuplösung** einer definierten Anzahl Verzeichnisse und Dateien in Powershell implementieren (genauere Vorgaben siehe unten)
 - 2) Das **Backup** soll eine Kopie der entsprechenden Verzeichnisse und Dateien sein.
 - 3) Es soll bei der Erstellung des Backups ein **Log** (bspw. Logdatei) erstellt und gespeichert werden.
 - 4) Nach Abschluss des Backups soll **überprüfbar** sein, ob der Backup **korrekt und vollständig** stattgefunden hat, durch überprüfen der Anzahl kopierter Objekte (Verzeichnisse und Dateien). Diese Überprüfung soll als Testfall definiert und dessen **Ergebnis im AA-Bericht** dargestellt werden.
 - 5) Es dürfen **keine «Drittkomponenten»** (wie bspw. Backuptools) verwendet werden.
 - 6) Falls sie **Bash** anstatt Powershell verwenden wollen, bitte mit Lehrperson Rücksprache nehmen.
- Als **Ergebnis** der **Anwendungsarbeit** soll folgendes entstehen (**AA-Bericht**)
- a) **Powershellskript** (eine oder mehrere ausführbare PS-Dateien), mit der Lösungsimplementation als Code
 - b) **PAP**, der den Ablauf ihrer Lösung darstellt
 - c) **Log** der Backup-Durchführung (kann Log-Datei sein)
 - d) Beschreibung von **Testfällen** für ihre Lösung, mit der sie die korrekte Funktion nachweisen; die Beschreibung der Testfälle muss das Erwartete **Ergebnis** und die verwendeten **Testdaten** je Testfall ebenfalls beinhalten.
 - e) **Durchführung** von Testfällen, Darstellung der Testergebnisse (ausreichend detailliert)

Zeitvorgaben und Abgaben (Finale und Zwischen-Abgaben), Video

- 1) Nicht-Einhaltung der Zeitvorgaben (Abgaben und Zwischenabgaben) gibt Bewertungsabzug.
Vorgaben: Siehe Tabelle unten.
- 2) - Neben dem **AA-Bericht** ist ein **Video** (3 Min. *) zu ihrer Lösung ihrer Anwendungsarbeit abzugeben, indem sie ausgewählte Kapitel aus dem AA-Bericht darstellen und erläutern.
- Die **Erläuterung** im Video (**mündlich gesprochener Kommentar** zum Video) hat in gleichen Teilen durch die Gruppenmitglieder zu erfolgen (bspw. indem sie eine Backupsequenz oder einen Testfall zeigen und mündlich Erläuterungen dazu abgeben). **Text-To-Speech** max. 1 Min.

PAP

Unten auf dem Bild ist unser PAP zu diesem Programm. Das PAP haben wir mit Visio gemacht und es zeigt das Vorgehen von unserem Programm. Nach der Ausführung soll man zuerst das zu speichernde Verzeichnis angeben, nachdem wir eine Eingabe gemacht haben und Enter gedrückt haben, soll man das Ziel Verzeichnis dann auch noch eingeben. Nach der Eingabe von dem Ziel Verzeichnis, gibt man «start-Backup» ein und die Funktion start-Backup wird durchgeführt.



Testfälle

1. Testfall

Im ersten Testfall haben wir getestet, ohne dass der C:\Backup Ordner existiert und geschaut, ob jener miterstellt wird beim ersten Implementieren. In den ersten paaren Versionen des Projekts funktionierte dies nicht immer und zum Teil funktionierte es, dann änderten wir bestimmte andere Parameter, welche die ursprünglichen in den Wahnsinn trieben. Schlussendlich haben wir es aber geschafft, dass der Ordner so erstellt wird.

2. Testfall

In der zweiten Test-Situation haben wir getestet, ob mehrere Backup-Ordner erstellt werden bei einem zweiten Start vom Skript. Dies konnte erfolgreich programmiert werden, da sich das Datum mit den Sekunden ständig ändert und somit der Name niemals gleichbleibt.

3. Testfall

Zu guter Letzt haben wir getestet, ob alle Dateien korrekt übertragen wurden. Das Ergebnis war positiv und wir mussten nicht allzu lange Verbesserungen vornehmen.

Bedienungsanleitung

1. Schritt

Das Skript wird über die PowerShell ISE geöffnet. Danach führt man das Skript über den grünen Startknopf oben aus.

2. Schritt

Anschliessend wird man gebeten zwei Eingaben zu machen. In der ersten Eingabe muss man das Verzeichnis nach folgendem Format eingeben, welches man speichern will.

Format: C:\Beispiel\Beispiel\Beispiel

3. Schritt

Danach wird man gebeten das Ziel-Verzeichnis anzugeben, in welches man das Verzeichnis abspeichern will. Das Format bei der Eingabe soll wie oben gezeigt sein.

4. Schritt

Um das Backup zu starten, muss man

Start-Backup

eingeben, um das Backup zu starten.

5. Schritt

Schlussendlich muss nur noch der Name des Log-Files angegeben werden und dann wird das Backup auch noch vollständig zu Ende geführt.

Fazit

Wir sind mit dem Ergebnis sehr zufrieden, auch wenn die Aufgabe nicht vollständig richtig gelöst werden konnte. Wir stiessen auf unvorhersehbare Probleme, welche nicht zeitgerecht ordentlich gelöst werden konnten. Ein Problem war, dass die Log-Datei nicht korrekt erstellt wurde, weil ich die Prozesse nicht tracen konnte. Jedoch steht sehr viel Arbeit und Lernprozess dahinter und das Skript wird in Zukunft privat noch vervollständigt und hoffentlich richtig zum Laufen gebracht. Die Recherche-Arbeit war sehr gross und es wurde man konnte nur sehr wenig vom Unterricht mitnehmen und anwenden. Man musste alles sich selbst beibringen und man konnte in keinem seiner vorherigen Skripts etwas gebrauchen ausser die Operatoren-Tabelle. Das Projekt musste innerhalb von knapp drei Wochen realisiert werden, welches viel zu wenig Zeit ist für ein derartig grosses Projekt. Das Projekt an sich wäre nicht so gross, jedoch ist die Arbeit, sich das alles selbst beizubringen, einen viel zu grossen Zeitaufwand und darum konnte das Projekt nicht vollständig den Richtlinien entsprechend abgegeben werden. Wir hoffen, dass bei diesen genannten Problemen Rücksicht genommen wird und auch fair bewertet wird, ich anhand des realisierten Projekts.

Anhänge

PowerShell Skript

```
# Legende
# Titel: Powershell-Backup
# Gruppe: Bernhard-Wang
# Datum/Version: 29.11.21
# Datei-Name: Version x

Write-Host ""
Write-Host ""
Write-Host ""
Write-Host ""

$TopSrc = Read-Host "Geben Sie das zu speichernde Verzeichnis an"
$TopBack = Read-Host "Geben Sie das Ziel-Verzeichnis an"

$AliasTopBack = $TopBack

Write-Host "Geben Sie 'start-Backup' ein"

#Globe Variablen

$Datum = get-date -Format dd-MM-yyyy_HH:mm:ss

function write-LogFile {
    param
    (
        [String]$StatusOK= "OK",
        [String]$StatusFail= "Failed",
        [String]$Event="CopyItem",
        [String]$Text,
        [String]$wichtig = $TopBack
    )

    if (!(Test-Path -Path ($AliasTopBack + "*\Log"))){
        $LogFolder = (New-Item -Path $AliasTopBack -name "Log" -ItemType Directory)
    }

    $LogFileNameC = Read-Host "Geben Sie den Namen für das LogFile ein: "

    # Erstellen des LogFiles
    $Logfile = New-Item -Path $LogFolder -Name ("$LogFileNameC" + "_" + "$Datum" +
    "_LogFile.txt") -ItemType File -Force

    # Überschrift für das LogFile
    Add-Content $Logfile ("Die LogDatei wurde erstellt am $(get-date -Format "dddd
    dd. MMMM yyyy HH:mm:ss") Uhr`n").ToUpper()

    # Leerzeilen einfügen
    Add-Content $Logfile "`n`n"

    # Spaltenüberschrift generieren
    $LogInhalt = "{0,-25}{1,-12}{2, -50}{3, -25}" -f
    "Zeit", "Event", "Daten", "Status"

    # Überschrift dem Logfile hinzufügen
    Add-Content $Logfile $LogInhalt

    # Generieren des Zeitstempels für die einzelnen Logzeilen
    $TimeStamp = get-date -Format "[dd.MM.yyyy HH:mm:ss]"

    # Inhalt ausgeben
    $LogInhalt = "{0,-25}{1,-12}{2, -50}{3, -25}" -f
    $TimeStamp, $Event, $TopSrc, $StatusOK

    # Hinzufügen zum LogFile
    Add-Content $Logfile $LogInhalt
}
```



```
Function start-Backup {  
    if (Test-Path -Path $TopBack){  
        $filename = "\Backup-"+ $Datum + "\"  
        $TopBack = New-Item -Path $TopBack -name $filename -ItemType Directory  
    }  
  
    else {  
        New-Item -Path $TopBack -ItemType Directory  
        $filename = "\Backup-"+ $Datum + "\"  
        $TopBack = New-Item -Path $TopBack -name $filename -ItemType Directory  
    }  
  
    Copy-Item $TopSrc -Recurse $TopBack -Force  
    $wichtig = $TopBack  
    write-LogFile($wichtig)  
}
```

Log-Datei

DIE LOGDATEI WURDE ERSTELLT AM MONTAG 13. DEZEMBER 2021 21:07:34 UHR

Zeit	Event	Daten	Status
[13.12.2021 21:07:34]	CopyItem	C:\TopSrc	D:\Backup\Backup-13-12-2021_210730

Dokumentationskonzept

a) Skriptfiles

- Projekt-Name
 - BackupSkript
- Datum
 - 13.12.2021
- File-Name
 - FinalVersion.ps1
- Versionsname
 - Final
- Globale Variablen
 - \$TopSrc
 - Quellverzeichnis
 - \$TopBack
 - Zielverzeichnis
 - \$AliasTopBack
 - \$TopBack wird hier auf eine andere Variable weiter festgelegt
 - \$Datum
 - Definiert das heutige Datum

b) Funktionen

- Write-LogFile {}
 - Zweck
 - Erstellt und schreibt in das Logfile
 - Verwendete Parameter
 - \$StatusOK
 - Gibt einen «OK» String aus
 - \$StatusFail
 - Gibt einen «Fail» String aus
 - \$Event
 - Gibt einen «CpyItem» String aus
 - \$Text
 - Noch nicht vollständig verwendet
 - \$Wichtig
 - Gibt den Wert von der start-Backup TopBack-Variable in die Funktion
 - Ausgaben
 - Gibt ein vollständiges LogFile.txt aus
- Start-Backup {}
 - Zweck
 - Startet das Backup mit der Unterfunktion
 - Ausgaben
 - Gibt einen Ordner mit dem vollständigen Backup aus beim gewünschten Verzeichnis