

Filtern von Informationen aus der Logdatei 'daemon.raw' - PowerShell Version

Unter Bezug auf meinen Tip *gener480.pdf* finden Sie umseitig das entsprechende Gerüst der Datei für eine Windows CMD Shell.

Hierzu ist anzumerken:

- In einem ersten Schritt werden alle Informationen für ein bestimmtes Datum extrahiert und in der Datei `All.txt` gespeichert
- Danach wird die Datei `All.txt` nach bestimmten Fehlern untersucht. Das Ergebnis wird in einer Datei entsprechenden Namens gespeichert. Auch wenn diese Datei leer ist, wird sie im Explorer mit der Größe von 1kB angezeigt. Aus diesem Grund muß dann mit einem Kniff eine Datei mit 0kB erst erzeugt werden
- Ist die Datei `0 kB` groß, ist die Meldung nicht aufgetreten.
- PowerShell kann mit Datumsangaben - auch Monats-übergreifend rechnen. Somit eignet sich das Script auch dazu, gleich die entsprechenden Unterverzeichnisse automatisch zu erstellen.
- Schließlich können Sie es damit in eine automatisch auszuführende Windows Task definieren.

Das Beispiel für ein CMD Gerüst finden Sie im Tip *windows_1.pdf*.

```

# Erstellen von nicht vorhandenen Verzeichnissen
#
$culture = New-Object System.Globalization.CultureInfo("en-US")
#
function touch {set-content -path ($args[0]) -value ($null) }
#
$daemon_raw_path = "D:\Legato\nsr\logs\daemon.raw"
$nsr_scripts_path = "D:\Legato\nsr\scripts"
$nsr_logs_path = "D:\Legato\nsr\logs"
$nsr_render_log = "D:\Legato\nsr\bin\nsr_render_log"
#
$datum = Get-Date
$datum = $datum.AddDays(-1)
$jahr = "{0:yyyy}" -f $datum
$monat = "{0:MM}" -f $datum
$tag = "{0:dd}" -f $datum
$Verzeichnis = "{0:MM_dd_yyyy}" -f $datum
$datum_us = "$monat" + "/" + "$tag" + "/" + "$jahr"
#
$path = $nsr_logs_path + "\" + $jahr
write $path
$result = test-path -path $path
write $result
if (!$result)
{
mkdir $path > null
}
#
# $path = $path + "\" + $jahr + "_" + $monat
$path = $path + "\" + $monat + "_" + $jahr
write $path
$result = test-path -path $path
write $result
if (!$result)
{
mkdir $path > null
}
#
.....

```

```

.....
#
$path = $path + "\" + $monat + "_" + $tag + "_" + $jahr
write $path
$result = test-path -path $path
write $result
if (!$result)
{
mkdir $path > null
}
#
.....
#
write "Step x/y"
$command = $nsr_render_log + " -S " + $datum_us + " -E " + $datum_us + " -achopty " + $daemon_raw_path + " > " //
+ $path + "\All.txt"
write $command
invoke-expression $command
#
cd $path
#
.....
#
write "Step x/y"
$lines = Get-Content buffer.txt
$lines | ForEach-Object {if ($_ -match "Cannot write to") {$ _ | Out-File 38758__Cannot_write_to.txt -append}}
$result = test-path -path 38758__Cannot_write_to.txt
if (!$result)
{
touch 38758__Cannot_write_to.txt
}
#
.....

```

```
.....
#
write "Step x/y"
$lines = Get-Content All.txt
$lines | ForEach-Object {if ($_ -match "39074")           {$_ | Out-File buffer.txt -append}}
$lines = Get-Content buffer.txt
$lines | ForEach-Object {if ($_ -match "Hardlimit")     {$_ | Out-File 39074__Hardlimit_reached.txt -append}}
$result = test-path -path 39074__Hardlimit_reached.txt
if (!$result)
{
touch 39074__Hardlimit_reached.txt
}
#
#
del buffer.txt
#
cd $nsr_scripts_path
#
write ""
write "done."
write ""
```