



Stapelverarbeitung mit ImageMagick

Description

Manchmal müssen viele Grafiken auf einmal konvertiert werden. Wie das in [Photoshop gemacht wird](#), haben wir bereits gezeigt. Doch Photoshop hilft nicht immer. An dieser Stelle kommt ImageMagick ins Spiel.

Die eierlegende Wollmilchsau

[ImageMagick](#) ist eine sehr mächtige Ansammlung von Tools. Der Schwerpunkt der freien Software ist die Erstellung und Bearbeitung von Pixelgrafiken. In diesem Tutorial konzentrieren wir uns auf den Konverter. Dieser ist so mächtig, dass man damit nahezu alles anstellen kann, was einem im Bildbereich in den Sinn kommt.

Das Problem für Anfänger: Es basiert auf Kommandozeilen und ist somit nicht ganz einfach zu bedienen, zumal [die dazugehörige Hilfe](#) dutzende von Parametern ausspuckt. Doch genau darin besteht auch die Macht des Konverters. Mit wenigen Zeilen kann man sich selbst eine Batch-Datei schreiben, die beliebig viele Bilder, wenn gewünscht auch mit Unterordnern, konvertiert.

Die Batch-Datei

Eine Batch-Datei (auch BAT-Datei genannt) ist im Prinzip eine Datei mit der Endung .bat, die ursprünglich unter DOS dazu diente, eine Reihe von Kommandozeilen abzuarbeiten. DOS-User, die noch die „guten alten Zeiten“ beschwören, werden sich an die berühmte *AUTOEXEC.BAT* erinnern. Sie diente unter DOS (und einigen alten Windows-Versionen) als Startdatei um beispielsweise Treiber beim Systemstart aufzurufen.

Selbst unter Windows 10 funktioniert diese Technik noch. So lassen sich heute bequem Stapelverarbeitungen erzeugen, die einem viel Arbeit abnehmen. Bei einer Stapelverarbeitung handelt es sich um wiederkehrende Aufgaben, die bis zu einem gewissen Endpunkt (zum Beispiel bis zur letzten Datei einer Liste) abgearbeitet werden.

Im Prinzip könnten wir auch unter Windows die Eingabeaufforderung mit `cmd.exe` starten und alles Zeile für Zeile eintippen, aber wir sind nicht doof und legen uns dafür eine Datei an.

Die Vorbereitung

Zunächst laden wir uns das oben verlinkte ImageMagick herunter und installieren es. Bei mir heißt der Ordner „ImageMagick-7.0.7-Q16“ und liegt auf der Partition **D**. Bei Dir kann der Ordner auch wo anders liegen, spielt zunächst keine Rolle. Im Ordner des Programms legen wir einen Unterordner mit dem Namen **!dummy** an.

Ordner mit diesen Namen benutze ich immer um temporär Dateien auszulagern und zu bearbeiten. Das hat den Vorteil, dass man das Original noch irgendwo hat, falls was schief läuft.

Im Beispiel dieses Tutorials wandeln wir eine beliebige Zahl von PNG Dateien in 32 Bit PNG um. Das ist nämlich ein Fall, den Photoshop nicht kann, wie auch viele andere Programme. Wenn eine Grafik keinen transparenten Bereich hat, wird sie von nahezu jedem Programm als 24 Bit gespeichert. 32 Bit kann aber nötig sein, wenn die beim Spiel verwendete Bibliothek einen Bug hat. Das Tutorial dient dabei nur als Beispiel, letztlich kann man, wenn das Grundgerüst steht, alles mögliche konvertieren. Dazu muss später nur eine Zeile angepasst werden.

Anschließend machen wir mit einem Texteditor eine neue Datei und speichern diese im ImageMagick-Ordner. Meine Datei heißt, aufgrund der Aufgabe, *png2png32.bat*. Wichtig ist hierbei die Endung *.bat*.

BAT-Code

Zum besseren Verständnis mache ich mehrere Abschnitte, der ganze folgende Code kommt aber in eine Datei. Der Kern wäre mit ein paar wenigen Zeilen abgehandelt, wir wollen dem kleinen Programm aber noch ein paar Features auf den Weg geben.

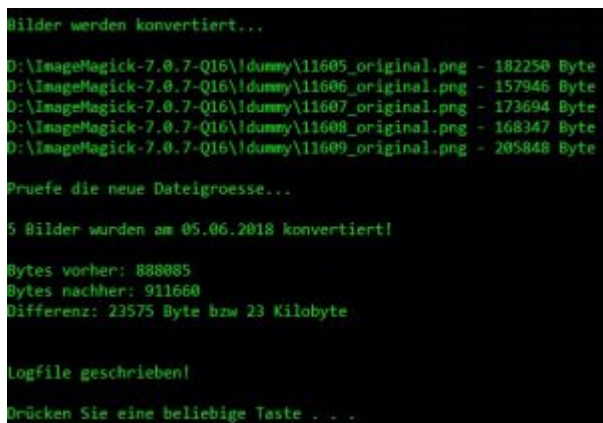
```
@echo off
SET folder="D:\ImageMagick-7.0.7-Q16\!dummy\"
SET convert="D:\ImageMagick-7.0.7-Q16\convert.exe"
SET /a logfile="logfile.txt"
SET /a sum=0
SET /a bytes_before=0
SET /a bytes_after=0
SET /a div=0
SET /a divkb=0
```

Mich *@echo off* verhindern wir, dass die Codezeilen auf dem Bildschirm ausgegeben werden. Anschließend definieren wir ein paar Variablen. Am wichtigsten sind *folder* und *convert*. Hier geben wir den Ordner an, in dem die Bilddateien liegen und wo der Konverter liegt.

Nun definieren wir mit *logfile* den Dateinamen einer Logile. Das hilft uns, einige Dinge zuprotokollieren. Wenn man sehr viele Bilder mit Unterordnern hat, dann ist es nach der Verarbeitung oft schwierig, den Überblick zu behalten. Die anderen Variablen brauchen wir, um ein paar Informationen zu erfassen und zu berechnen.

```
for /R %folder% %%f in (*.png) do (
    echo %%f - %%~zf Byte
    echo %%f - %%~zf Byte >> logfile
    SET /a bytes_before+=%%~zf
    %convert% %%f -define png:format=png32 %%f
    SET /a sum+=1
)
```

Das ist der wesentliche Teil des Programms. Es ist eine [for-Schleife](#), in der alle Bilder im Ordner *folder* abgearbeitet werden. Mit dem Parameter */R* geht das Programm alle Unterverzeichnisse durch. *%%f* ist die Bilddatei. Dabei schreiben wir noch Dateiinformationen in die Logfile und zählen mit *SET /a sum+=1* wie viele Dateien bereits bearbeitet wurden. Wer auf den ganzen Schnickschnack verzichten möchte, kann sich auf die wesentliche Zeile der Schleife konzentrieren:



```
Bilder werden konvertiert...
D:\ImageMagick-7.0.7-Q16\dummy\11605_original.png - 182250 Byte
D:\ImageMagick-7.0.7-Q16\dummy\11606_original.png - 157946 Byte
D:\ImageMagick-7.0.7-Q16\dummy\11607_original.png - 173694 Byte
D:\ImageMagick-7.0.7-Q16\dummy\11608_original.png - 168347 Byte
D:\ImageMagick-7.0.7-Q16\dummy\11609_original.png - 205848 Byte

Pruefe die neue Dateigroesse...

5 Bilder wurden am 05.06.2018 konvertiert!

Bytes vorher: 888885
Bytes nachher: 911660
Differenz: 23575 Byte bzw 23 Kilobyte

Logfile geschrieben!

Drücken Sie eine beliebige Taste . . .
```

%convert% ist die eingangs definierte Variable für den Konverter. In *%%f* steckt der Name der zu konvertierenden Datei. Anschließend folgt der Befehl *-define png:format=png32*. Den kannst Du nun frei definieren, je nachdem, was der Konverter tun soll. *%%f* am Ende ist die Ausgabedatei. Wir überschreiben also einfach die alte Datei.

Ausgeschrieben könnte das auch so lauten:

```
D:\ImageMagick-7.0.7-Q16\convert.exe D:\ImageMagick-7.0.7-Q16\!dummy\bild.png
```

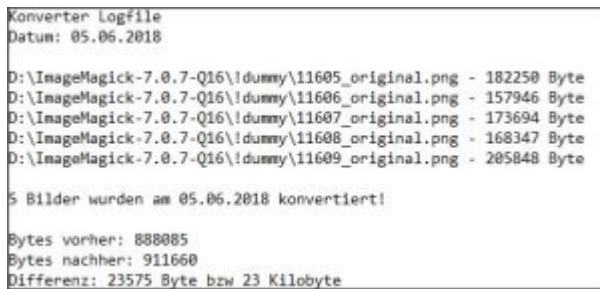
In der Schleife prüfen wir auch, wie groß eine Datei ist und zählen das zusammen.

```
echo.
echo %sum% Bilder wurden am %date% konvertiert!
echo.
echo Bytes vorher: %bytes_before%
```

```
echo Bytes nachher: %bytes_after%
echo Differenz: %div% Byte bzw %divkb% Kilobyte
echo.

echo. >> logfile
echo %sum% Bilder wurden am %date% konvertiert! >> logfile
echo. >> logfile
echo Bytes vorher: %bytes_before% >> logfile
echo Bytes nachher: %bytes_after% >> logfile
echo Differenz: %div% Byte bzw %divkb% Kilobyte >> logfile

echo.
echo Logfile geschrieben!
echo.
pause
```



```
Konverter Logfile
Datum: 05.06.2018

D:\ImageMagick-7.0.7-Q16\dummy\11605_original.png - 182250 Byte
D:\ImageMagick-7.0.7-Q16\dummy\11606_original.png - 157946 Byte
D:\ImageMagick-7.0.7-Q16\dummy\11607_original.png - 173694 Byte
D:\ImageMagick-7.0.7-Q16\dummy\11608_original.png - 168347 Byte
D:\ImageMagick-7.0.7-Q16\dummy\11609_original.png - 205848 Byte

5 Bilder wurden am 05.06.2018 konvertiert!

Bytes vorher: 888085
Bytes nachher: 911660
Differenz: 23575 Byte bzw 23 Kilobyte
```

Das *pause*

am Ende verhindert nur, dass sich das Fenster gleich wieder schließt.

Zugegeben, es ist für einen, der das noch nie gemacht hat, viel Information. [D sollte sich unbedingt diese Seite anschauen](#). Die beiden Screenshots zeigen zudem, was man auf dem Bildschirm und in der L

Date Created

5. Juni 2018

Author

sven