

Automatisiertes Backup mit Hilfe einer Batchdatei unter Windows

angefertigt von
Dominik Erdmann
www.der-domi.de

Januar 2006

Vorwort

Um einem Datenverlust vorzubeugen, ist es wichtig regelmäßig eine Sicherung seiner Daten zu machen. Bisher habe ich dies durch viel Handarbeit erreicht.

Da ich nicht die ganze Zeit neben dem Rechner sitzen und warten möchte bis der nächste Schritt erledigt ist, habe mir eine Lösung überlegt. Mit einer BATCH-Datei läßt sich unter Windows ein Ablauf automatisieren. Dieses möchte ich euch nicht vorenthalten.

Die BATCH-Datei erstellt aus einem Verzeichnis ein Archiv. Da das Archiv oft sehr groß ist, wird es in mehrere Teile gespalten. So lassen sich die einzelnen Dateien auf eine CD oder DVD kopieren.

Aus eigener Erfahrung habe ich festgestellt, dass die Komprimierung mit bzip2 besser und schneller ist, wie das windowseigene zip.

Benutzen kannst du es auch, wenn du viele Dateien transportieren willst und kein genügend großes Speichermedium zu Verfügung hast.

Viel Spass,
Dominik

Inhaltsverzeichnis

1	Grundsätzliches	1
1.1	Batchdateien	1
1.2	Tar	1
1.3	Bzip2	1
1.4	Programmdateien vorbereiten	2
1.4.1	Möglichkeit 1: Integration der Dateien durch Systemvariable	2
1.4.2	Möglichkeit 2: Integration durch temporäres Ändern	4
2	Sicherung erstellen und zurückspielen	5
2.1	Batchdateien vorbereiten	5
2.1.1	Setting - Datei	5
2.1.2	Backup starten - Datei	6
2.1.3	Backup zurückspielen - Datei	7
2.2	Backup starten	8
2.3	Backup zurückspielen	9

Kapitel 1

Grundsätzliches

1.1 Batchdateien

Mit Batchdateien, die auch als Batchprogramme (Stapelverarbeitungsprogramme) oder Skripts bezeichnet werden, können Routinetasks oder sich ständig wiederholende Tasks vereinfacht werden. Eine Batchdatei ist eine nicht formatierte Textdatei, die einen oder mehrere Befehle enthält und die Dateinamenerweiterung `.bat` oder `.cmd` hat. [batch]

1.2 Tar

tar ist der Name eines im Unix-Umfeld sehr geläufigen Archivierungsprogramms. Außerdem wird so auch das Dateiformat bezeichnet, welches von diesem Programm verwendet wird. Der Name wurde aus tape archiver (Bandarchivierer) gebildet, da mit dem Programm ursprünglich Daten auf Bandlaufwerken gesichert wurden. Gleichzeitig ist tar auch das englische Wort für Teer (mit dem Programm werden Dateien unkomprimiert zu einer Datei zusammengeklebt).

Tar bietet die Möglichkeit, Dateien sequenziell in eine einzige Datei zu schreiben, bzw. Dateien aus selbiger wieder herzustellen. Die entstehende Datei trägt die Endung `.tar` (unkomprimiert) und wird auch als Tarball (dt. Teerklumpen oder Teerkugel) bezeichnet. [tar]

1.3 Bzip2

bzip2 ist ein frei verfügbares Komprimierungsprogramm zur verlustfreien Kompression von Dateien, entwickelt von Julian Seward. Es ist frei von jeglichen patentierten Algorithmen und wird unter einer BSD-ähnlichen Lizenz vertrieben.

bzip arbeitet mit dem Burrows-Wheeler Block-Sorting-Algorithmus, der Text zwar nicht komprimiert, aber besser komprimierbar macht (Präkodierung), sowie mit Huffman-Kodierung.

Die Kompression mit bzip2 ist in den meisten Fällen effektiver, aber auch langsamer als die Kompression mit gzip. Mit bzip2 komprimierte Dateien werden durch die Dateiendung `.bz2` gekennzeichnet. TAR-Dateien, die mit bzip2 komprimiert wurden, haben üblicherweise die Erweiterung `.tar.bz2` oder auch `.tbz`.

bzip2 ist der Nachfolger von bzip, das ursprünglich arithmetisches Kodieren nach dem Blocksort benutzte; aus patentrechtlichen Gründen wurde bzip jedoch nicht mehr weiterentwickelt. [bzip2]

1.4 Programmdateien vorbereiten

Zur Sicherungserstellung wird die Datei `backup.bat` verwendet. `backupRecover.bat` spielt die Sicherung wieder zurück.

Während der Sicherung werden verschiedene Programmaufrufe durchgeführt. Da diese Programme von Windows nicht bereitgestellt werden, müssen sie beim Ausführen der Batchdateien bekannt sein. Bei den Programmen handelt es sich um

- `bzip2.exe`,
- `md5sum.exe`,
- `split.exe` und
- `tar.exe`¹.

Die Programme sind in dem Paket enthalten (Verzeichnis `bin`) und unterliegen der GNU Public License [GPL].

Am einfachsten ist es, wenn die Programme ins System integriert wird. So muss man sich nicht mehr darum kümmern, dass die Dateien vorhanden sind.

1.4.1 Möglichkeit 1: Integration der Dateien durch Systemvariable

Dazu kopiert man die Programme in einen neuen Ordner beliebiger Wahl. Den Ordernamen kopiert man, wie Abbildung 1.1 gezeigt.

Unter `Start->Einstellungen->Systemsteuerung->System` wird der Ordner als Systemvariable bekannt gegeben (siehe Abbildung 1.2). Füge ans Ende der Zeile ein Semikolon `;` und dann den kopierten Ordner. Die Vorgehensweise kann u.a. etwas anders aussehen.

¹`tar.exe` erfordert `libiconv-2.dll` und `libintl-2.dll`

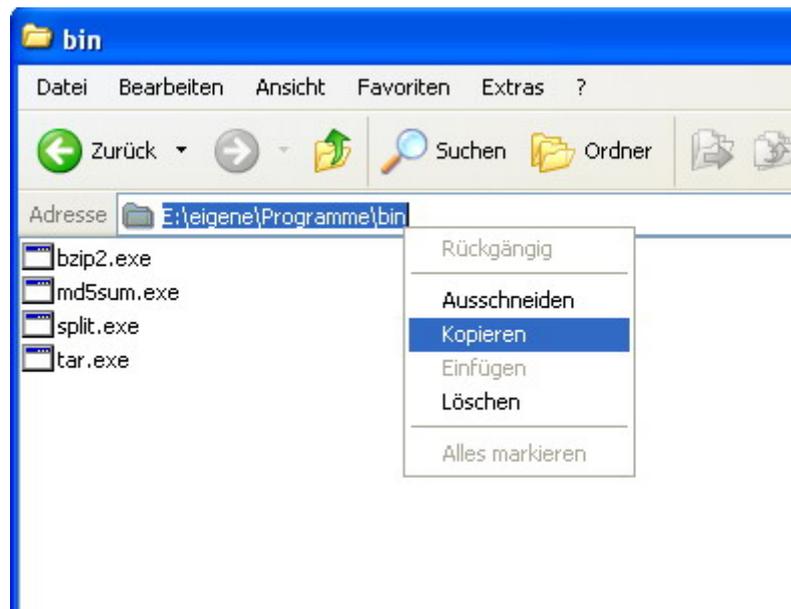


Abbildung 1.1: Programmfenster

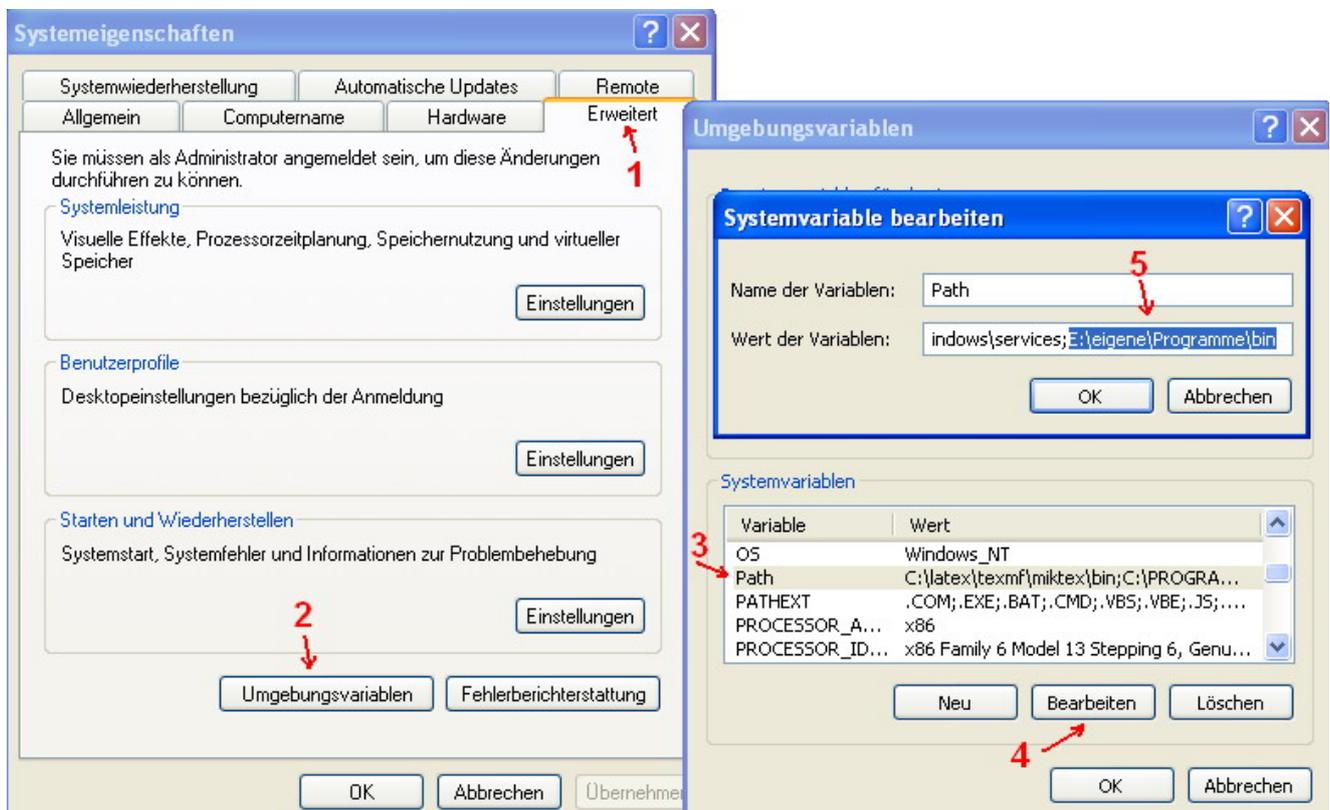


Abbildung 1.2: Integration des Ordners ins System

1.4.2 Möglichkeit 2: Integration durch temporäres Ändern

Siehe dazu in Listing 2.1, Zeile 18. Dies ist die einfachere Möglichkeit. Möglichkeit 1 ist aus „historischen“ Gründen noch vorhanden.

Kapitel 2

Sicherung erstellen und zurückspielen

Damit die Batchprogramm weiß, welcher Ordner gesichert werden soll, müssen Variablen gesetzt werden. Dazu kann die Batchdatei durch einen Rechtsklick zum Bearbeiten geöffnet werden.

2.1 Batchdateien vorbereiten

2.1.1 Setting - Datei

Listing 2.1: backup_settings.bat

```
1 rem :: Formatiert date nach JJJJ-MM-TT;
2 rem kommt es zu Fehlern, dann Zeile auskommentieren
3 for /F "tokens=1,2,3,4 delims=/. " %%a in ('date/T') do set date=%%c-%%b-%%a
4
5 rem :: Verzeichnis, welches gesichert wird
6 set source_path=E:\eigene
7
8 rem :: Zieldateiname fuer Archiv
9 set destination_file=backup_%date%
10
11 rem :: Zeilspeicherpfad fuer Archiv
12 set destination_path=E:\backup
13
14 rem :: Dateigroesse in MB des Archivs; wenn 0, dann nicht splitten
15 set destination_size=700
16
17 rem :: Verzeichnis, welches die Programmdateien enthaelt
18 set programmfiles=E:\eigene\Programme\bin
```

Zeile 6 Die Variable `source_path` gibt den Pfad an, welcher gesichert werden soll.

Zeile 9 Mit `destination_file` wird festgelegt, wie die Dateien nach der Sicherung heißen. Durch `%date%` wird das aktuelle Datum in der Form JJJJ-MM-TT in den Namen geschrieben. Sollte

es mit dieser Variable Probleme geben oder das Datum in der Form TT.MM.JJJJ gewünscht, kann Zeile 3 auskommentiert werden, indem `rem` vorangeschrieben wird.

Zeile 12 In `destination_path` wird das Backup gespeichert. Hier müssen die Verzeichnisse mit `\` getrennt werden.

Zeile 15 `destination_size` gibt an, welche Größe in Megabyte die Dateien haben wird. Gebe 0 an, wenn das Archiv nicht gesplittet werden soll.

Zeile 18 Die Variable `programmfiles` gibt an, in welchem Ordner die Programmdateien liegen. Falls man nach Möglichkeit 1 wie in Kapitel 1.4.1 vorgeht, bleibt die Variable leer.

Zeile 1 Sollte während dem Backup ein Fehler auftreten, kann diese Zeile auskommentiert werden. Dadurch sieht man alle Ausgaben auf der Kommandozeile.

2.1.2 Backup starten - Datei

Listing 2.2: backup_start.bat

```
1 @echo off
2
3 rem :: speichere Anfangszeit zur Ausgabe am Schluss
4 set start=%time%
5
6 rem :: aufrufen der Settingdatei
7 call backup_settings.bat
8
9 rem :: Verzeichnis zu Programmdateien bekannt geben
10 path=%path%;"%programmfiles%"
11
12 rem :: erstelle Pfad, falls nicht vorhanden
13 mkdir "%destination_path%"
14
15 rem :: wechsle in den Zielpfad
16 cd "%destination_path%"
17
18 rem :: packe den Ordner
19 tar -cf "%destination_file%.tar" "%source_path%"
20
21 rem :: komprimiere das Archiv
22 bzip2 -zf "%destination_file%.tar"
23
24 rem :: erstelle Checksumme des Archivs, zur Integritaetpruefung
25 md5sum "%destination_file%.tar.bz2" >> "%destination_file%_tar_bz2_md5sum.txt"
26
27 rem :: spalte Archiv in mehrere kleinere Dateien
28 if not %destination_size%==0 split -b%destination_size%m "%destination_file%.tar.bz2" "%destination_file%."
```

```

29
30 rem :: loesche temporaere Datei
31 if not %destination_size%==0 del "%destination_file%.tar.bz2"
32
33 rem :: erstelle Checksumme der gesplitteten Dateien, zur Integritaetpruefung
34 if not %destination_size%==0 (
35     for %%f in ("%destination_file%.*") do (
36         md5sum "%%f" >> "%destination_file%_md5sum.txt"
37     )
38 )
39
40 rem :: Info zur Dauer ausgeben
41 echo Datensicherung begann um %start% und endete um %time%
42
43 rem :: warten, damit Hinweise gelesen werden koennen
44 pause

```

Zeile 1 Sollte während dem Backup ein Fehler auftreten, kann diese Zeile auskommentiert werden. Dadurch sieht man alle Ausgaben auf der Kommandozeile.

Zeile 7 Hier wird der Name der Setting-Datei eingegeben. Sollte sie in einem anderen Verzeichnis liegen, muss der Pfad mit angegeben werden.

2.1.3 Backup zurückspielen - Datei

Listing 2.3: backup_recover.bat

```

1 @echo off
2
3 rem :: speichere Anfangszeit zur Ausgabe am Schluss
4 set start=%time%
5
6 rem :: aufrufen der Settingdatei
7 call backup_settings.bat
8
9 rem :: Verzeichnis zu Programmdateien bekannt geben
10 path=%path%;"%programfiles%"
11
12 rem :: wechsle in Verzeichnis mit Backupdateien
13 cd "%destination_path%"
14
15 rem :: falls Backup nicht gesplittet wurde, existiert das Archiv bereits
16 if %destination_size%==0 goto notSplitted
17
18 rem :: vergleiche Archiv mit Checksumme, um Datenintegritaet festzustellen
19 md5sum -c "%destination_file%_md5sum.txt"
20

```

```
21 rem :: fuege gespalteten Dateien wieder zusammen
22 copy /b "%destination_file%.*" "%destination_file%.tar.bz2"
23
24 rem :: Sprungmarke
25 :notSplitted
26
27 rem :: vergleiche Archiv mit Checksumme, um Datenintegritaet festzustellen
28 md5sum -c "%destination_file%_tar_bz2_md5sum.txt"
29
30 rem :: Hinweis ausgeben
31 echo Schaeue, ob alle Checksummen OK sind!
32
33 rem :: warten, um Checksumme pruefen zu lassen
34 pause
35
36 rem :: verschiebe Archiv in oberstes Verzeichnis
37 move "%destination_file%.tar.bz2" \
38
39 rem :: wechsle ins oberste Verzeichnis
40 cd \
41
42 rem :: dekomprimiere Archiv und entpacken
43 bzip2 -cd "%destination_file%.tar.bz2" | tar -xpf -
44
45 rem :: loesche Archiv, falls Archiv gesplittet
46 if not %destination_size%==0 del "%destination_file%.tar.bz2"
47
48 rem :: verschiebe Archiv, falls Archiv nicht gesplittet
49 if %destination_size%==0 move "%destination_file%.tar.bz2" "%backup_path%"
50
51 rem :: Info zur Dauer ausgeben
52 echo Recovery begann um %start% und endete um %time%
53
54 rem :: fertig
55 pause
```

Zeile 1 Sollte während dem Recovery (Backup zurückspielen) ein Fehler auftreten, kann diese Zeile auskommentiert werden. Dadurch sieht man alle Ausgaben auf der Kommandozeile.

Zeile 7 Hier wird der Name der Setting-Datei eingegeben. Sollte sie in einem anderen Verzeichnis liegen, muss der Pfad mit angegeben werden.

2.2 Backup starten

Nun kann `backup_start.bat` von jedem Verzeichnis durch einen Doppelklick gestartet werden. Achte darauf, dass die Batchdatei immer au dem gleichen Verzeichnis liegen.

2.3 Backup zurückspielen

Beim Zurückspielen des Backups müssen die Archivdateien auf dem gewünschten Ziellaufwerk liegen, da die Laufwerksinformation im Archiv nicht vorhanden ist.

Nun kann `backup_recover.bat` von jedem Verzeichnis durch einen Doppelklick gestartet werden.

ACHTUNG

Beim Zurückspielen werden vorhandene, neuere Dateien überschrieben.

Checksumme

Beim Zurückspielen werden die Dateien auf ihre Datenintegrität¹ geprüft. Listing 2.4 zeigt, wie soetwas aussehen könnte.

Listing 2.4: Beispiel eines Zurückspielen

```
1 E:\backup>backupRecovery.bat
2 backup_2006-01-15.aa: OK
3 backup_2006-01-15.ab: OK
4 backup_2006-01-15.ac: OK
5 backup_2006-01-15.ad: OK
6 backup_2006-01-15.ae: OK
7 backup_2006-01-15.aa
8 backup_2006-01-15.ab
9 backup_2006-01-15.ac
10 backup_2006-01-15.ad
11 backup_2006-01-15.ae
12      5 Datei(en) kopiert.
13 backup_2006-01-15.tar.bz2: OK
14 Schaue, ob Checksummen OK sind!
15 Drücken Sie eine beliebige Taste . . .
```

Zeile 2 bis 6 Hier werden die gesplitteten Dateien überprüft. Kommt es zu Fehlern, müssen die Dateien nochmals kopiert werden. Achte darauf, dass das Archiv (Endung `.tar.bz2`) gelöscht werden muss.

Zeile 7 bis 12 Die Dateien werden zusammengefügt. Sollte das Archiv bereits existieren, werden Zeile 2 bis 12 übersprungen.

Zeile 13 Die Archivdatei wird überprüft.

¹Unter Datenintegrität versteht man die logische Korrektheit, Gültigkeit und Genauigkeit von Daten.

Literaturverzeichnis

[batch] *Verwenden von Batchdateien*; Hilfe- und Supportcenter, Windows XP Professional

[bzip2] <http://de.wikipedia.org/wiki/Bzip2>

[GPL] <http://www.gnu.org/copyleft/gpl.html>

[tar] <http://de.wikipedia.org/wiki/Tar>