

BildUmb

Benennt beliebig viele Bild-(oder andere)dateien in einem Arbeitsgang um.

Sollen Fotos in einer Fotogalerie im Internet oder auf einer CD veröffentlicht werden, müssen die Dateinamen häufig bestimmten Namenskonventionen genügen, z. B. um mit JavaScript-Funktionen effizienter verwaltet werden zu können. Hat man viele Dateien umzubenennen, ist das eine eintönige und ermüdende Arbeit, die überdies Fehlerquellen birgt.

Hier schafft das Programm **BildUmb** Abhilfe: Es kann viele hundert Dateien in einem oder wenigen Arbeitsgängen umbenennen. (Tip: Auch zum Umbenennen von anderen Dateien kann man es „mißbrauchen“ ([S. U.](#))). Man kann es in zwei Betriebsarten benutzen:

- Im **Kommandozeilenmodus**, auf den sich diese Anleitung bezieht, gibt man sämtliche nötigen Angaben zusammen mit dem Befehl in der Befehlszeile an.
- Im **interaktiven Modus** fragt das Programm die Angaben vom Benutzer ab. Es erklärt sich im wesentlichen selbst; bei Unklarheiten über die Begrifflichkeiten lohnt sich aber auch hier die Lektüre dieser Anleitung.

Man gelangt in den interaktiven Modus, indem man das Programm einfach im Dateibetrachter (Windows-Explorer, Konqueror, Dolphin, ...) anklickt oder im Befehlsfenster BildUmb eingibt.

Einige Punkte sind jedoch auch hier zu beachten:

- Wird BildUmb durch Anklicken im Dateibetrachter gestartet, muß die Programmdatei Bildumb.exe (Windows/DOS) bzw. BildUmb (Linux) im interaktiven Modus im selben Verzeichnis liegen wie die umzubenennenden Dateien.
- Dateinamen mit Leerzeichen oder Umlauten können unter Windows **nur** im interaktiven Modus umbenannt werden. In diesem Fall müssen die Eingaben in Anführungszeichen eingeschlossen werden.

Anleitung für die Benutzung im Kommandozeilenmodus:

Syntax:

```
BildUmb <Präfix_alt> <Präfix_neu> <Nr_Anfang> <Nr_Ende> [<Stellen_alt>  
[<Stellen_neu> [<Verschiebung> [<Suffix_alt> [<Suffix_neu>]]]]]
```

Angaben in eckigen Klammern können weggelassen werden, jedoch müssen bei Verwendung eines dieser Parameter auch alle vorherigen angegeben werden.

Was damit gemeint ist, erklärt folgendes Anwendungsbeispiel:

Gabi will an einen Kollegen mehrere Digitalfotos von einem Betriebsausflug schicken. Ihre Digitalkamera versieht diese mit Dateinamen der Form:

IMG_101_007.JPG

Für die Bildergalerie im Intranet sollen sie jedoch in der Form

Gabi-1234.jpg

vorliegen; außerdem sollen die Bilder mit aufeinanderfolgenden Nummern versehen sein, und schließlich sollen noch kleine Vorschaubilder erzeugt werden nach dem Muster:

Gabi-1234k.jpg

Gabi hat keine Lust, die Bilder von Hand umzubenennen und bedient sich des Programms **BildUmb**. Das Programm arbeitet in zwei Stufen:

1. Zunächst wird anhand der Benutzereingaben eine Stapelverarbeitungsdatei erzeugt (also eine Datei mit Betriebssystembefehlen, die für Windows/DOS und Linux unterschiedlich sind). Diese enthält die eigentlichen Umbenennungsbefehle.
2. Im zweiten Schritt werden dann mit Hilfe dieser Datei die einzelnen Bilddateien umbenannt. Sollte man bei der Eingabe einen Fehler gemacht haben, kann nicht viel passieren, denn der Versuch, nicht vorhandene Dateien umzubenennen, führt lediglich zu einer Fehlermeldung des Betriebssystems; das Gleiche gilt, wenn versucht wird, eine vorhandene Datei in eine andere vorhandene Datei umzubenennen: In beiden Fällen findet keine Umbenennung statt.

Gabi hat auf dem Ausflug 30 Fotos gemacht. Diese tragen in ihrer Kamera die Namen IMG_101_081.JPG bis IMG_101_110.JPG. Einige Fotos (92 bis 95) sind jedoch nichts geworden, so daß sie sie nicht weitergeben will.

1. Um die Fotos umzubenennen, legt sie auf ihrem Rechner als erstes ein leeres Verzeichnis an und kopiert die Bilddateien dort hinein. Das Programm **BildUmb** wird ebenfalls in diesem Verzeichnis abgelegt (oder in einem anderen Verzeichnis, das im Suchpfad vorkommt).
2. Danach öffnet sie ein Befehlsfenster:
 - Windows 95/98: Programme -> Zubehör -> MS-DOS-Eingabeaufforderung oder Windows-Taste -> Ausführen: command oder command.com eingeben.
 - Windows NT/2000/XP: Programme -> Zubehör -> Eingabeaufforderung oder Windows-Taste -> Ausführen: cmd oder cmd.exe, am besten mit der Option /f
 - Linux: bash starten, je nach Distribution und Benutzeroberfläche unterschiedlich, aber F2 drücken und bash eingeben müßte immer funktionieren.

3. Mit dem Befehl `cd` wechselt sie in das Verzeichnis, in dem die Bilder liegen:

- Windows/DOS z. B.:

```
cd "C:\Dokumente und Einstellungen\Gabi\Eigene Dateien\Bilder\Betriebsausflug"
```

 (Die Anführungszeichen sind notwendig, weil im Verzeichnisnamen Leerzeichen vorkommen.)
- Linux z. B.:

```
cd ~/Bilder/Betriebsausflug
```

Mit `cd ..` wechselt man vom aktuellen Verzeichnis in das darüberliegende.

`\` (Windows/DOS) bzw. `/` (Linux) bezeichnet das Hauptverzeichnis des jeweiligen Laufwerks bzw. der gesamten Dateihierarchie.

4. Als nächstes muß sie die Stapelverarbeitungsdatei erzeugen. Dazu gibt sie gemäß folgender Syntax ein:

```
BildUmb <Präfix_alt> <Präfix_neu> <Nr_Anfang> <Nr_Ende> [<Stellen_alt>
[<Stellen_neu> [<Verschiebung> [<Suffix_alt> [<Suffix_neu>]]]]]
```

Angaben in eckigen Klammern können weggelassen werden, jedoch müssen bei Verwendung eines dieser Parameter auch alle vorherigen angegeben werden. Wenn Gabi also z. B. eine Verschiebung wünscht (s. u.), muß sie auch Angaben zu Stellen_alt und Stellen_neu machen.

Dabei ist:

- **Präfix_alt**: Der Teil am Anfang des Dateinamens, der bei allen umzubenennenden Dateien gleich ist. In unserem Beispiel ist das `IMG_101_`.
- **Präfix_neu**: Der Teil des Dateinamens, der nach der Umbenennung vor dem Nummernanteil stehen soll. Als neuer Präfix wäre also `Gabi-` anzugeben.
- **Nr_Anfang** und **Nr_Ende**: Die Nummern des ersten und letzten Bildes, das umbenannt werden soll. In unserem Beispiel hat Gabi die mißlungenen Bilder 92 bis 95 gar nicht erst in das Verzeichnis kopiert. Die entsprechenden Dateinamen sind also nicht vorhanden. Das schadet jedoch nicht.
- **Stellen_alt** und **Stellen_neu**: Bei Gabi liegen die Bildnummern dreistellig mit führenden Nullen vor (001 bis 999). Denkbar wären auch zwei (01 - 99) oder vier Stellen (0001 - 9999) oder eine Numerierung ohne führende Nullen (hat die Zahl mehr Stellen, als vorgesehen sind, wird der Dateiname entsprechend länger). Die Numerierung ohne führende Nullen führt jedoch im Betriebssystem zu einer falschen Sortierung: Nach 1 kommen 11 ... 19, dann erst die Nummer 2. Daher führt **BildUmb** immer eine Umbenennung **mit** führenden Nullen durch, es sei denn, es wird für **Stellen_neu** 0 oder 1 angegeben. Der Benutzer kann sich jedoch aussuchen, wieviele Stellen die Numerierung haben soll, insbesondere kann er angeben, mit wievielen Stellen die bisherigen Dateinamen vorliegen. **BildUmb** berücksichtigt bei der Umbenennung führende Nullen nach den Vorgaben des Benutzers:
Läßt man die Angabe für **Stellen_alt** weg, nimmt **BildUmb** an, daß die Bilder ohne führende Nullen numeriert sind.
Wird für **Stellen_neu** nichts angegeben, erfolgt die Umbenennung vierstellig.
Seit Programmversion 4.0 ist der Benutzer völlig frei bei der Angabe der Stellenzahl. (Vorher konnte nur von 1 bis 4 bzw. 2 bis 4 gewählt werden.)
- **Verschiebung**: Dies wird benötigt, wenn man neben den Textanteilen auch die Nummern ändern will. Der Kollege hat Gabi gebeten, die Bilder mit fortlaufenden Nummern zu versehen, d. h., daß keine Lücken in dem Nummernintervall vorkommen sollen. Die Nummern 96 bis 110 sollen also um 4 Zähler auf 92 bis 106 verschoben werden.
- **Suffix_alt** und **Suffix_neu**: Suffix ist der Teil des Dateinamens, der nach der Numerierung, aber vor der Dateierdung (.jpg) kommt. Bei Gabis Bildern kommt hier gar nichts, bei den kleinen Vorschaubildern ist „k“ der Suffix. Die eckigen Klammern im Syntaxschema bedeuten, daß die Teile darin auch weggelassen werden können. Da der neue Suffix nochmals in einer weiteren eckigen Klammer eingeklammert ist, kann man diesen weglassen, während man den alten Suffix angibt (die neuen Dateinamen hätten dann also keinen Suffix mehr), nicht jedoch umgekehrt. Einen leeren Suffix wird durch zwei Anführungszeichen "" dargestellt.
Man kann also
 - einen alten und einen neuen Suffix angeben,
 - nur den alten Suffix angeben (der neue wird dann als leer angenommen) oder
 - beide weglassen, wenn kein Suffix vorhanden und gewünscht wird.In unserem Beispiel trifft letzteres zu.
Achtung: Gelegentlich wurde berichtet, daß in der Linux-Version eine Stapelverarbeitungsdatei mit nicht definierten Dateinamen erzeugt wird, falls keine Suffixe angegeben werden. In diesem Fall gibt man einfach zwei leere Suffixe an: "" "".

Da manche Bildnummern verschoben werden müssen, muß Gabi das Programm zweimal ausführen:

- Für die Bilder 81 bis 91 gibt sie ein:

```
BildUmb IMG_101_ Gabi- 81 91 3 4 0 "" ""
```

Die Teile dieser Befehlszeile bedeuten also, daß Gabi

- die Bilder 81 bis 91 umbenennen will,
- die dreistellig mit führenden Nullen vorliegen („3“)
- und später vierstellig numeriert sein sollen („4“).
- Die Bildnummer soll nicht verändert werden („0“),
- und weder vor noch nach der Umbenennung haben die Dateinamen einen Suffix („"" und """).

Im allgemeinen können die Anführungszeichen (= leere Suffixe) am Ende auch weggelassen werden.

- Da die Bilder 92 bis 95 fehlen, müssen die darüberliegenden verschoben werden:

```
BildUmb IMG_101_ Gabi- 96 110 3 4 -4 "" ""
```

Mit der Verschiebung -4 wird also das Bild 96 zu 92 usw.

- Bei den kleinen Vorschaubildern nach dem Schema IMG_101_007k.JPG würde das letzte Beispiel folgendermaßen lauten:

```
BildUmb IMG_101_ Gabi- 96 110 -4 k k
```

Der Suffix „k“ bleibt hier also unverändert.

Jeweils nach der Eingabe des Befehls muß nun im Verzeichns eine Datei namens BildU.bat (Windows/DOS) bzw. BildU (Linux) stehen. Im Beispiel würde die Datei nach den jeweiligen Durchgängen so aussehen:

Bildnummern	Windows/DOS	Linux
81 - 92	<pre>@echo off ren IMG_101_081.* Gabi-0081.jpg ren IMG_101_082.* Gabi-0082.jpg ren IMG_101_083.* Gabi-0083.jpg ren IMG_101_084.* Gabi-0084.jpg ren IMG_101_085.* Gabi-0085.jpg ren IMG_101_086.* Gabi-0086.jpg ren IMG_101_087.* Gabi-0087.jpg ren IMG_101_088.* Gabi-0088.jpg ren IMG_101_089.* Gabi-0089.jpg ren IMG_101_090.* Gabi-0090.jpg ren IMG_101_091.* Gabi-0091.jpg</pre>	<pre>#!/bin/bash mv -i IMG_101_081.* Gabi-0081.jpg mv -i IMG_101_082.* Gabi-0082.jpg mv -i IMG_101_083.* Gabi-0083.jpg mv -i IMG_101_084.* Gabi-0084.jpg mv -i IMG_101_085.* Gabi-0085.jpg mv -i IMG_101_086.* Gabi-0086.jpg mv -i IMG_101_087.* Gabi-0087.jpg mv -i IMG_101_088.* Gabi-0088.jpg mv -i IMG_101_089.* Gabi-0089.jpg mv -i IMG_101_090.* Gabi-0090.jpg mv -i IMG_101_091.* Gabi-0091.jpg</pre>

5. Im letzten Schritt wird die soeben erzeugte Stapelverarbeitungsdatei ausgeführt:

- Unter Windows/DOS gibt es die Befehlszeile, in der man die Datei BildU.bat anführt. (Groß-/Kleinschreibung wird ignoriert), oder sie klickt im Windows-Explorer die Datei BildU.bat an.
 - Unter Linux muß vorher noch ein Punkt gesetzt werden: `. BildU` oder `./BildU`.
- Hier kommen es auf Groß- und Kleinschreibung an, aber mit der Befehlsergänzung (Anfang des Befehls eingeben und die Tab-Taste drücken) kommt man dennoch viel schneller ans Ziel. Alternativ kann auch hier die Datei im Konqueror angeklickt werden.

Danach sind die Dateien im Verzeichnis, in dem die Stapelverarbeitungsdatei erzeugt wurde, und „Ausführung“ müssen jedoch für jeden Durchgang erneut ausgeführt werden, da die Stapelverarbeitungsdatei jeweils überschrieben wird:

Also zunächst die Datei BildU.bat bzw. BildU für die Bildnummern 81 bis 91 erzeugen, dann ausführen. Dann sind die Bilder 81 bis 91 umbenannt. Danach die Schritte für die Nummern

96 bis 110 mit der leicht veränderten Befehlszeile wiederholen.

Tip: Unter Windows NT und höher und unter Linux kann man im Befehlsfenster mit der „Pfeil-nach-oben“-Taste die vorhergehenden Kommandozeilenbefehle wieder aufrufen, ggf. verändern und erneut ausführen. So spart man sich viel Schreibarbeit bei der wiederholten Verwendung gleicher oder ähnlicher Befehlszeilen wie in unserem Beispiel. Unter DOS und Windows 95/98 kann man zumindest mit der „Pfeil-nach-rechts“-Taste den vorigen Befehl Zeichen für Zeichen wieder hervorholen.

6. Am Ende kann die Stapelverarbeitungsdatei gelöscht werden.

7. **Tip:** Mit **BildUmb** kann man nicht nur Bilddateien mit der Endung .jpg umbenennen, sondern grundsätzlich jeden Dateityp. Zwar haben nach der Umbenennung alle Dateien die Endung .jpg, aber mit einem einfachen Befehl kann man allen .jpg-Dateien in einem Verzeichnis auf einmal eine andere Endung geben. Haben Ihre Dateien z. B. die Endung .html, geben Sie unter Windows/DOS einfach ein:

```
ren *.jpg *.html
```

Danach haben alle Dateien, die vorher mit .jpg endeten, die Endung .html (auch die, die gar nicht an der Umbenennung beteiligt waren; daher bietet sich dieses Verfahren nur an, wenn keine solchen Dateien im Verzeichnis liegen). Statt .html kann natürlich auch jede andere Endung verwendet werden.

Programm herunterladen:

<http://EDVBK.de/Programme/>

letzte Änderung: 02.02.2012