

Batch-Programmierung

Wikibooks

Bibliografische Information

Detaillierte Daten zu dieser Publikation sind bei Wikibooks zu erhalten:

`HTTP://DE.WIKIBOOKS.ORG/`

Namen von Programmen und Produkten sowie sonstige Angaben sind häufig geschützt. Da es auch freie Bezeichnungen gibt, wird das Symbol ® nicht verwendet.

Erstellt am 9. März 2011.

Diese Publikation ist entstanden bei Wikibooks, einem Projekt für Lehr-, Sach- und Fachbücher unter den Lizenzen Creative Commons Attribution/Share-Alike (CC-BY-SA) und GFDL.

PDF- und Druckversion sind entstanden mit dem Programm `wb2pdf` unter GPL. Dabei wurde das Textsatzprogramm `LATEX` verwendet, das unter der LPPL steht.

Einzelheiten und Quellen dazu stehen im Anhang.

Inhaltsverzeichnis

1	ALLGEMEINE INFORMATION	3
1.1	BATCH PROGRAMMIERUNG	3
1.2	WIE ERSTELLE ICH EINE BATCHDATEI?	4
1.3	ALLGEMEINE INFORMATIONEN	5
2	WICHTIGE DOS-KOMMANDOS	7
2.1	ASSOC	7
2.2	ATTRIB	7
2.3	CACLS	8
2.4	CD / CHDIR	8
2.5	CLS	10
2.6	COMP ODER FC	10
2.7	(X)COPY/ROBOCOPY	10
2.8	DATE	11
2.9	DEL / ERASE	12
2.10	DELTREE	12
2.11	DIR	13
2.12	EDIT	14
2.13	EXIT	14
2.14	FDISK	14
2.15	FIND	14
2.16	FORMAT	15
2.17	HELP	15
2.18	IPCONFIG	15
2.19	KEYB	16
2.20	LOADHIGH	16

2.21	MD	16
2.22	MORE	16
2.23	MOVE	17
2.24	NETSH	17
2.25	PATH	18
2.26	PROMPT	18
2.27	RD / RMDIR	18
2.28	REN ODER RENAME	18
2.29	START	19
2.30	SET	19
2.31	SETLOCAL	19
2.32	TIME	19
2.33	TYPE	20
2.34	TITLE	20
2.35	TASKKILL	20
3	BATCH BEFEHLE	21
3.1	@	21
3.2	:(DOPPELPUNKT)	22
3.3	CALL	23
3.4	CLS	25
3.5	CMD	26
3.6	COLOR	27
3.7	COMMAND	28
3.8	ECHO	28
3.9	FOR	31
3.10	GOTO	36
3.11	IF	36
3.12	PAUSE	38
3.13	REM	40
3.14	START	40
3.15	VARIABLEN	42
3.16	TASKKILL	45

4	BATCH OPERATOREN	47
4.1	&	47
4.2	47
4.3	&&	48
4.4	49
4.5	<	49
4.6	>	50
4.7	>>	51
5	ERWEITERUNGEN UNTER WINDOWS NT	53
5.1	BATCHPARAMETER	53
5.2	STRINGMANIPULATION	54
5.3	UMGEBUNGSVARIABLEN	55
5.4	DIE KOMMANDOZEILE LERNT RECHNEN	57
5.5	BENUTZEREINGABEN	58
5.6	AUSGABEN IN EINE VARIABLE UMLEITEN	59
6	PROGRAMMIERUNGSHILFEN	61
7	NETZWERKUMGEBUNG	77
7.1	PING	77
7.2	IPCONFIG	78
7.3	TRACERT	81
7.4	NETSTAT	82
7.5	NET	82
8	BEISPIELE	89
9	ZUSATZ-TOOLS	109
10	REFERENZ	111
10.1	LISTE ALLER INTERNEN UND EXTERNEN BEFEHLE (ENTSPRICHT DER AUSGABE DES BEFEHLS help UNTER WINDOWS XP)	111

11 WEBLINKS	117
11.1 WEBLINKS	117
12 AUTOREN	121
13 BILDNACHWEIS	127

KATEGORIE:BUCH MIT DRUCKVERSION¹

1 [HTTP://DE.WIKIBOOKS.ORG/WIKI/KATEGORIE%3ABUCH%20MIT%20DRUCKVERSION](http://de.wikibooks.org/wiki/Kategorie%3ABuch%20mit%20Druckversion)

1 Allgemeine Information

1.1 Batch Programmierung

Batchprogramme (häufig mit Stapelverarbeitungsprogramme oder kurz Stapelprogramm übersetzt) sind meist kurze Dateien, die Befehle der Kommandozeile der Reihe nach abarbeiten. Batchprogrammierung erfüllt die grundlegenden Anforderungen an eine Programmiersprache wie z.B. `if`, `if not` und `while`. Da Batch sich in den Bereich Shellscripting einordnen lässt, kann man von einer *interpreterbasierten Skriptsprache* reden. Das heißt, dass die im Editor erstellte Textdatei nicht mit einem COMPILER¹ einmal in Maschinensprache übersetzt und in einer ausführbaren Binärdatei im `*.exe` Format gespeichert, sondern bei jedem Aufruf durch einen Interpreter zur Laufzeit in ein für den Computer verständliches Format übersetzt wird. Ein Texteditor wie MS Notepad, den Windows von Haus aus mitbringt, reicht vollkommen aus um Batchprogramme zu schreiben. Batchprogrammierung ist speziell für die Steuerung des Betriebssystems gedacht, für die Entwicklung von Anwendungen ist sie nicht geeignet. Hierzu verwendet man Sprachen wie C/C++ und Java (nicht zu verwechseln mit *JavaScript!*).

Viele wiederkehrende Installations- und Verwaltungsaufgaben, die man traditionell mit Tastatur und Maus ausführt, lassen

¹ [HTTP://DE.WIKIBOOKS.ORG/WIKI/COMPILER](http://de.wikibooks.org/wiki/Compiler)

sich mit Stapeldateien ausführen. Hat man genügend Kommentarzeilen in die Batchdatei eingefügt, genügt der Ausdruck der Stapeldatei meist als Dokumentation.

Kommandozeilenbefehle und Batchdateien sind keine veraltete Technologie. Auf Wunsch vieler Systemadministratoren hat Microsoft dafür gesorgt, dass Windows Server 2003 vollständig von der Kommandozeile installiert und administriert werden kann, ohne die Maus zu benutzen.

In diesem Buch geht es ausschließlich um die Erstellung von Batch-Dateien unter Windows und DOS. Andere Betriebssysteme bieten vergleichbare und wesentlich mächtigere Funktionen (wie z. B. die in der Linux-Welt weit verbreitete Bash oder AppleScript für den Mac).

Die Begriffe "Stapelprogramm" und "Stapelverarbeitung" rühren aus der Zeit her, als Permanentenspeicher auf Magnetplatten sehr teuer war. Zu dieser Zeit begab man sich mit einem Stapel Lochkarten aus Pappe, auf denen sich das Programm und die Daten in gestanzter Form befanden, zum Rechenzentrum und übergab dem Operator seinen Stapel. Die Stapel aller Benutzer wurden gesammelt und dann sequentiell an den Rechner übergeben. Nach einiger Zeit konnte man sich dann die (meist ausgedruckten) Ergebnisse abholen.

1.2 Wie erstelle ich eine Batchdatei?

Wie bereits gesagt, reicht ein einfacher Editor aus. Empfehlen kann man guten Gewissens den quelloffenen und kostenlosen NOTEPAD++²-Editor, als bessere Alternative zu MS Notepad. Speichern Sie dann die Datei mit der Endung *.bat ab. Anschließend

2 [HTTP://NOTEPAD-PLUS.SOURCEFORGE.NET/DE/SITE.HTM](http://notepad-plus.sourceforge.net/de/site.htm)

müssen Sie nur die Datei starten: Dies können Sie entweder direkt in Windows oder indem Sie in der DOS-Eingabeaufforderung in das entsprechende Verzeichnis wechseln und den Dateinamen eingeben.

Unter Windows NT-kompatiblen Betriebssystemen gibt es seit Windows 2000 auch Batchdateien mit der Endung *.cmd. Diese werden genau wie Batchdateien mit der Endung *.bat verarbeitet bzw. ausgeführt. Die Endung *.cmd wurde ursprünglich aus Kompatibilitätsgründen zu OS/2, einem Betriebssystem das MS ursprünglich in Kooperation mit IBM entwickelte, mit Windows NT 3.x eingeführt.

1.3 Allgemeine Informationen

- Allgemeine Informationen: WIKIPEDIA: BATCH³
- Infos zur cmd.exe: WIKIPEDIA: CMD.EXE⁴
- Die für **Windows XP** verfügbaren Kommandos können auch im Hilfesystem von Windows direkt eingesehen werden:
 - START - Hilfe und Support
 - "Befehlszeilenreferenz A-Z" in das Suchfeld eintragen. (Für englisches Windows "Command-line reference A-Z")

3 [HTTP://DE.WIKIPEDIA.ORG/WIKI/STAPELVERARBEITUNG](http://de.wikipedia.org/wiki/Stapelverarbeitung)

4 [HTTP://DE.WIKIPEDIA.ORG/WIKI/CMD.EXE](http://de.wikipedia.org/wiki/Cmd.exe)

2 Wichtige DOS-Kommandos

Man unterscheidet bei DOS-Befehlen grundsätzlich zwischen internen und externen Befehlen. Interne Befehle sind Bestandteil der Shell selbst, während externe Befehle als eigene Programme vorliegen. Die DOS-Befehle, die interessant für die Batch-Programmierung sein könnten, sind im Folgenden aufgelistet:

2.1 assoc

Mit Hilfe dieses Befehls kann man die Dateierweiterungen bearbeiten bzw. anzeigen lassen.

```
assoc [.dateiendung[=Dateibesreibung]]
```

2.2 attrib

Ändert das Dateiattribut. Möglich sind "schreibgeschützt" (+r/-r), "versteckt" (+h/-h) und "archiv" (+a/-a)

2.3 cacls

Ändert die Rechte (Access Control List) auf NTFS Partitionen oder zeigt diese an.

```
cacls *.* /E /T /G Bob:C
```

Fügt beispielsweise dem User Bob das Recht "Ändern" für alle Dateien und Ordner im aktuellen und allen Unterordnern hinzu. **Achtung!** Standardmäßig werden Rechte ersetzt!

2.4 cd / chdir

Über den `cd`- oder `chdir`-Befehl (change directory) kann das aktuelle Verzeichnis gewechselt werden. Beachte: Nach dem "cd" folgt ein Leerzeichen, das bei nachfolgendem `..` oder `\` je nach Betriebssystemversion weggelassen werden darf. Wird der Befehl ohne Parameter ausgeführt, so wird der Pfad des aktuellen Verzeichnisses ausgegeben.

relative Angaben:

Die Eingabe von "`cd ..`" wechselt in das nächsthöhere Verzeichnis ("`cd ..\..`" zwei Ebenen höher usw.).

Die Eingabe von "`cd ORDNERNAME`" wechselt in den Unterordner `ORDNERNAME`.

absolute Angaben:

Die Eingabe von "cd \" wechselt in das Hauptverzeichnis auf dem derzeitig befindlichen Laufwerk.

Die Eingabe von "cd \ORDNERNAME" wechselt absolut in den Ordner ORDNERNAME auf dem derzeitig befindlichen Laufwerk.

Die Eingabe von "LAUFWERKSBUCHSTABE:" (ohne vorangestelltes "cd") wechselt in das aktuelle Verzeichnis des gewählten Laufwerks LAUFWERKSBUCHSTABE.

Die Eingabe von "cd LAUFWERKSBUCHSTABE:\ORDNERNAME" wechselt absolut in den Ordner ORDNERNAME auf dem gewählten Laufwerk von LAUFWERKSBUCHSTABE jedoch nicht von einem Laufwerk auf ein anderes.

Pfadangaben mit Leerschlägen (Bsp "C:\program files\") müssen in Anführungs- /Schlusszeichen eingeschlossen sein.

Der Befehl "cd /D %~dp0" wechselt in den Ordner des Batch-Skripts. (sinnvoll bei Windows-Doppelklicks)

Wildcards:

Ist die Pfadangabe eindeutig kann der Ordnername durch Nutzen von Wildcards (z.b. \win* für \windows oder \wintnt) abgekürzt werden. Dies ist jedoch im Batchbetrieb nicht ratsam.

Remotesysteme:

Pfade von nicht als Laufwerken verbundenen Remotesystemen können (Berechtigung vorausgesetzt) entweder über die Administrative Freigabe oder durch das Verbinden eines Netzlaufwerks angesprochen werden. Beispiel für administrative Freigabe: \\[computername]\[laufwerk]\$\[Weiterer Pfad]

Pfad in Variable:

Der Befehl "set PFAD=%cd%" schreibt das aktuelle Verzeichnis in die Variable %PFAD%. (als letztes im Batch-Script mit "cd %PFAD%" wieder ins Startverzeichnis wechseln!)

2.5 cls

Löscht den Bildschirminhalt (**clear screen**)

2.6 comp oder fc

Vergleicht den Inhalt zweier Dateien und stellt die Unterschiede dar (**compare / file compare**).

2.7 (x)copy/robocopy

Mit dem Befehl `copy` können eine Datei oder mehrere Dateien kopiert werden.

`xcopy` ist die erweiterte Version von `copy`, die mehr Optionen bietet. Diese ist nicht in allen Windows-Versionen verfügbar.

Das weitaus mächtigere `robocopy` ist ab Vista standardmäßig wieder verfügbar, bei Windows XP und 2000 fehlt dieses Programm zunächst. Microsoft stellt `robocopy` aber im Windows Server 2003 Resource Kit Tools 2003 auch für diese Betriebssystemversionen bereit.

Mit dem Befehl `copy /b "DATEI" + "DATEI" "NEUE DATEI"` kann man Dateien zusammenfügen.

2.8 date

Gibt das aktuelle Datum aus und ermöglicht dem Benutzer die Änderung des Datums. Wird der Befehl mit dem Parameter `/t` aufgerufen, so wird nur das aktuelle Datum ausgegeben. Um das Datum zu teilen, also z.B. nur das Jahr zu verwenden, gibt es folgende Syntax:

```
date:~6,4
```

Beispiel:

```
z:\>echo %date:~6,4%  
2010
```

Hier werden vom Datum die ersten 6 Zeichen weggelassen und dann vier Stellen angezeigt. Bei `TIME`¹ funktioniert das analog.

¹ [HTTP://DE.WIKIBOOKS.ORG/WIKI/%23TIME](http://de.wikibooks.org/wiki/%23TIME)

Bei einigen Betriebssystemversionen erfordert das Ändern des Systemdatums administrative Rechte.

XP: Die Ausgabe des Datumsformates ist abhängig von den Einstellungen in den Regions- und Sprachoptionen (Systemsteuerung)

2.9 del / erase

Löscht eine Datei, die sich im aktuellen Verzeichnis befindet. Als Parameter muss mindestens der Dateiname angegeben werden. Weitere Parameter sind optional:

- /p - sollen mehrere Dateien gelöscht werden, muss jede einzelne Löschung bestätigt werden.
- /f - auch schreibgeschützte Dateien werden gelöscht.
- /s - löscht auch Dateien, die sich in den Unterverzeichnissen befinden.
- /q - Unterbindet Bestätigung.

Beispiel:

```
erase H:\zulöschendedatei.bat
```

2.10 deltree

Löscht Verzeichnis(se) inkl. untergeordnete Dateien.

Unter Windows XP lautet der Befehl **rmdir**. Deltree kennt es nicht mehr. Anmerkung: rmdir/rd versteht keine Wildcards ('*' oder '?') im Verzeichnisnamen.

2.11 dir

Der Befehl `dir` (directory) zeigt alle Verzeichnisse und Unterverzeichnisse an. Verzeichnisse werden seit Windows 95 als Ordner bezeichnet, unterscheiden sich allerdings nicht von Verzeichnissen.

Der `dir`-Befehl hat zahlreiche Optionen:

- `/b` - Nur der Dateiname wird ausgegeben. Die Ausgabe aller weiteren Informationen wie beispielsweise Datum, Größe oder Datenträgerbezeichnung wird unterdrückt.
- `/c` - Die Dateigröße wird mit einem Punkt als Trennzeichen zwischen jeweils drei Dezimalstellen angegeben, also beispielsweise 343.232 (Standardeinstellung). Soll dies unterdrückt werden, geben Sie als Option `/-c` an.
- `/p` - Bei jedem Seitenumbruch wird die Ausgabe unterbrochen, bis der Anwender eine Taste drückt.
- `/on` - Gibt das Verzeichnis alphabetisch sortiert aus.
- `/w` - Mehrere Einträge in einer Zeile werden angezeigt.
- `/ah` - Versteckte Dateien werden angezeigt.
- `/a` - Alle versteckten- und System-Dateien werden angezeigt.
- `/s` - Zeigt auch den Inhalt der Unterverzeichnisse an.

Mit dem Befehl `dir [Dateiname]` ist es auch möglich, Dateien im aktuellen Verzeichnis zu suchen. Alternativ kann auch der Platzhalter `*` (bspw.: `dir *.*`) verwendet werden, um nach bestimmten Dateien zu suchen.

Mit `dir *` werden nur Objekte ohne Dateinamen-Erweiterung angezeigt. In der Regel haben Ordner keine Erweiterung und Dateien eine Erweiterung. In diesem Fall werden nur Ordner angezeigt. Ordner werden im DOS mit `<dir>` vor dem Ordnernamen gekennzeichnet.

2.11.1 Kombinationen

`dir /w/o/s/p` durchsucht den ganzen Rechner nach einer Datei und stellt das Ergebnis seitenweise dar.

2.12 edit

Mit `edit` wird ein Editor geöffnet, zum Beispiel zum Bearbeiten von `BATCH`²- oder Textdateien.

2.13 exit

Beendet die Eingabeaufforderung/kehrt zu Windows zurück.

2.14 fdisk

Partitioniert die Festplatte.

2.15 find

Mit dem Befehl `find` kann - auch in mehreren Dateien - nach einer Zeichenfolge gesucht werden.

Auch `find` kann mit Parametern gesteuert werden:

- `find /i` ignoriert Groß-/Kleinschreibung beim Suchbegriff
- `find /v` Zeigt alle Zeilen an, die die Zeichenfolge NICHT enthalten.

² [HTTP://DE.WIKIBOOKS.ORG/WIKI/BATCH](http://de.wikibooks.org/wiki/Batch)

- `find /c` Zeigt nur die Anzahl der die Zeichenfolge enthaltenden Zeilen an.
- `find /n` Zeigt die Zeilen mit ihren Zeilennummern an.

2.16 format

Formatiert die Festplatte oder Diskette. Mit `format c:` wird die Festplatte `c:` formatiert. Mit `format a:` wird die Diskette im Laufwerk `a:` formatiert. Mit `format c: -s` bzw. `format a: -s` kann man die Festplatte/Diskette mit Systemdateien formatieren, so das sie nachher bootfähig sind.

2.17 help

Die Hilfe-Funktion steht in allen Windows-Systemen (außer Win9X, d. h. 95/98/ME) zur Verfügung. Der allgemeine Befehl hierzu lautet `help`. Als Ausgabe wird eine Liste mit einem Teil der verfügbaren Kommandozeilen-Befehle ausgegeben. Diese Liste umfasst den Befehlsnamen (linke Spalte) und eine Beschreibung (rechte Spalte). Soll ein Befehl genauer erklärt werden, so lautet die Eingabe `help BEFEHLSNAME` oder auch `BEFEHLSNAME /?`. Diese Eingabe in die Kommandozeile fördert detaillierte Informationen und z.T. auch Beispiele zu einem Befehl zu Tage. Es werden jedoch nicht alle Befehle die es im DOS gibt angezeigt.

2.18 ipconfig

Informationen über die Windows-IP-Konfiguration

2.19 keyb

Lädt Tastaturtreiber. `keyb gr,c:\dos\keyboard.sys` lädt den deutschen Tastaturtreiber (`keyboard.sys` muss sich im Verzeichnis `c:\dos\` befinden)

2.20 loadhigh

Lädt Programme in freie UMB Upper-Memory-Blocks nicht bei xp verfügbare

2.21 md

Der Befehl `md` (**make directory**) legt ein neues Verzeichnis an.

2.22 more

Mit der Ausgabeumleitung `| more` (bspw.: `type [Dateiname] | more`) wird bei Seitenwechsel zur Bestätigung aufgefordert.

Damit besteht die Möglichkeit über mehrere Seiten führende Dokumente anzuzeigen.

`more` ist der einzige Befehl, bei der auch in der DOS-Welt die sogenannte Pipe verwendet wird. Bei der Pipe handelt es sich um das Zeichen `|`. Mit diesem Zeichen ist es möglich, zwei Befehle miteinander auf eine spezielle Weise zu verbinden. Dabei wird der erste Befehl ausgeführt, seine Ausgaben werden jedoch nicht auf den Bildschirm geschrieben, sondern in einen Puffer. Der zweite

Befehl wird auf eine Weise gestartet, dass er seine Eingaben nicht von der Tastatur, sondern von eben diesem Puffer bekommt.

2.23 move

Verschieben von einer oder mehrerer Dateien von einem Verzeichnis in ein anderes.

oder

ein Verzeichnis umzubenennen

```
move test test_neu
```

2.24 netsh

Ohne Parametereingabe ergibt sich ein neuer Kontext. Weitere Hilfe mit: ?

Beispiel : netsh interface dump > ipconf1.dat

diese Zeile als Batch-Datei ausgeführt, legt die momentane IP Konfiguration in einer Datei namens: ipconf1.dat ab ! Umgekehrt kann aus einer Datei z.B. namens ipconf2.dat eine neue IP Konfiguration geladen und übernommen werden und zwar so:

```
netsh -f ipconf2.dat
```

So lässt sich eine IP Konfiguration ohne Umherclicken in Systemmenues ablegen und (modifiziert) wieder zurückholen.

2.25 path

Durch die Angabe eines Pfades wird das Betriebssystem auf der Kommandozeile dazu aufgefordert, neben dem aktuellen Verzeichnis auch in weiteren Verzeichnissen zu suchen. Wird der Befehl ohne Parameter ausgeführt, so werden alle aktuellen Pfade ausgegeben.

2.26 prompt

Legt das Aussehen der Eingabezeile fest.

2.27 rd / rmdir

Der Befehl `rd` oder `rmdir` (remove directory) löscht ein Verzeichnis, wenn dieses leer ist.

Parameter:

- `/s` Löscht einen ganzen Verzeichnisbaum mit Dateien
- `/q` Löscht ohne Nachfrage ganze Verzeichnisbäume. Kann in Batch-Dateien praktisch sein.

Der Verzeichnisname darf keine Wildcards enthalten (*' oder '?').

Beispiel: `rmdir /s /q C:\temp`

2.28 ren oder rename

Datei(en) umbenennen.

2.29 start

Öffnet neues Fenster für eigenes Programm.

Beispiel: (WinXP) zum öffnen eines neuen Konsolenfensters in dem eine Datei aufgerufen wird.

```
start "Name des neuen Fensters" /WAIT /D C:\temp /NORMAL  
call test.bat
```

/WAIT = Wartet bis Anwendung geschlossen wird.

/NORMAL = Startet Anwendung in der NORMAL-Prioritätsklasse.

/D = Danach folgt das Startverzeichnis

2.30 set

`set` setzt eine temporäre Umgebungsvariable oder gibt die Variablenwerte aus. Temporär bedeutet jeweils nur für die aktuelle cmd Session.

2.31 setlocal

`setlocal` Steuert die Begrenzung des Gültigkeitsbereiches von Änderungen. Wird mit `endlocal` beendet.

2.32 time

Gibt die aktuelle Zeit aus und ermöglicht dem Benutzer die Änderung der Uhrzeit.

Bei einigen Betriebssystemversionen erfordert das Ändern des Systemdatums administrative Rechte.

2.33 type

Gibt den (reinen Text-)Inhalt der angegebenen Datei (z. B. *.bat, *.cmd oder *.txt) aus.

2.34 title

Dieser Befehl ändert die Fensterüberschrift des Programmfensters.

```
title Beispiel
```

2.35 taskkill

Ab Windows XP kann man mit diesem Befehl einen Prozess beenden, beispielsweise wenn dieser nicht mehr reagiert. Unter Einbindung von /t werden auch alle untergeordneten Prozesse beendet, und wenn man /f hinzufügt, wird das Beenden erzwungen.

```
taskkill /f iexplore.exe /t
```

3 Batch Befehle

3.1 @

Schaltet die Ausgabe der Befehlszeile auf dem Bildschirm nur für den aktuellen Befehl aus und ist selbst kein eigener Befehl.

Syntax

```
@befehl
```

Stapelanweisung: Ab MS-DOS bis Windows 7

Beispiel:

Inhalt echo Diese Zeile wird mit Befehlszeile ausgeführt...

```
@echo und diese ohne!
```

Ausgabe C:\>echo Diese Zeile wird mit Befehlszeile ausgeführt...

```
Diese Zeile wird mit Befehlszeile ausgeführt...
```

```
und diese ohne!
```

In Batch Files verhindert "@echo off" zu Beginn des Skriptes die Ausgabe aller (!) Befehlszeilen auf dem Bildschirm bis die Stapelverarbeitung beendet wird, abbricht oder mittendrin ein "echo on" Befehl erfolgt, um z. B. Befehlszeilen tatsächlich anzuzeigen und dann auszuführen. Kommentare (mit ::¹ oder

¹ [HTTP://DE.WIKIBOOKS.ORG/WIKI/%23%3A%20%28DOPPELPUNKT%29](http://de.wikibooks.org/wiki/%23%3A%20%28DOPPELPUNKT%29)

REM²) werden natürlich auch nicht angezeigt. Ist aber nur ein Nebeneffekt. @echo off

REM Verhindert, dass dieser Kommentar angezeigt wird.

Ohne @echo off: C:\>REM Dieser Kommentar wird so angezeigt

3.2 : (Doppelpunkt)

Sprungmarke für ein Unterprogramm bzw. eine Kommentarzeile.

Sprungmarken werden benötigt, wenn mittels der Batchdatei eine Bedingung überprüft und erfüllt bzw nicht erfüllt wird und entsprechend weiter verfahren werden soll.

Mit dem Batchbefehl goto wird die Sprungmarke angesprungen.

Anmerkung:

Der Doppelpunkt hat auch die Funktion der Manipulation von Variablen, wenn er direkt hinter einer Variablen steht. Siehe Kapitel "Variablen" in dieser Publikation.

Syntax

:NAMEDERSPRUNGMARKE

Sprungmarken können eine beliebige Länge haben, von cmd behandelt werden allerdings nur die ersten 8 Zeichen, der Rest wird ignoriert. Kommen in einer Batch also :Sprungmarke1 und

2 [HTTP://DE.WIKIBOOKS.ORG/WIKI/%23REM](http://de.wikibooks.org/wiki/%23REM)

:Sprungmarke2 vor, so wird nur die erste beim Aufruf einer der Beiden gefunden.

Beispiel:

```
Inhalt if exist C:\blabla.txt goto EDITBLA
goto END
:: Kommentarzeile, sofern es erforderlich ist, einen Kommentar zu
schreiben
:: Zur Unterscheidung von Sprungmarken verwende ich zwei "::"
:EDITBLA
edit blabla.txt
:END
```

Sofern die Datei C:\blabla.txt existiert, wird sie mit edit geöffnet, sonst wird das Unterprogramm übersprungen und die Batchdatei bei der Marke :END fortgesetzt, also beendet.

Seit die Befehlsweiterungen aktiviert sind, steht in Batchdateien die Sprungmarke :EOF zur Verfügung, welche sich unsichtbar am Ende der Batch-Datei befindet.

3.3 CALL

Mit call kann man eine andere Batch-Datei aufrufen. Sobald diese beendet wurde, wird die ursprüngliche Batchdatei weiter ausgeführt.

```
Beispiel: REM Call.cmd
@echo off
echo Diese Batchdatei ruft eine andere auf.
call anderedatei.bat
pause
```

```
REM anderedatei.bat
@echo off
echo Die andere Datei
```

Die Ausgabe wenn man die Datei Call.cmd startet:

```
Diese Batchdatei ruft eine andere auf.
Die andere Datei
{Pause}
```

Wenn die Befehlsweiterungen aktiviert sind (Standard ab Windows 2000) kann man auch Sprungmarken aufrufen (und auch Parameter übergeben):

```
Beispiel: @echo off
REM Diese Batchdatei ruft eine eigene Sprungmarke auf
echo Vor dem Sprung
call :sprungmarke
echo Nach dem Sprung
pause
goto end
:sprungmarke
echo Sprungmarke aufgerufen!
goto :eof
:end
exit
```

Die Ausgabe: Vor dem Sprung

```
Sprungmarke aufgerufen!
Nach dem Sprung
```

Beispiel mit Parameter: @echo off

```
REM Diese Batchdatei ruft eine eigene Sprungmarke auf
```

```
echo Vor dem Sprung
call :sprungmarke meinParameter
echo Nach dem Sprung
pause
goto end
:sprungmarke
echo Sprungmarke aufgerufen und Parameter %1 uebergeben!
goto :eof
:end
exit
```

Die Ausgabe: Vor dem Sprung

Sprungmarke aufgerufen und Parameter meinParameter uebergeben!

Nach dem Sprung

```
goto :eof
```

Dieser Befehl springt automatisch zum Ende der Batchdatei
(beendet die aktuelle Prozedur)

3.4 CLS

Mit `cls` (clear screen) wird der Bildschirm gelöscht.

Syntax

```
cls
```

Interner Befehl: Ab MS-DOS bis Windows 7

Beispiel:

Inhalt @echo off

```
echo Hier schreibe ich jetzt ganz viel Text.
```

```
echo Hier kann ich z.B. hinschreiben, dass ich jemanden mag.  
echo.  
echo Aber den Text sieht man gleich sowieso nicht mehr... Hihi!  
cls  
echo War irgendetwas?  
pause>nul
```

Ausgabe War irgendetwas?

3.5 CMD

Das Programm, das es möglich macht, DOS-Befehle auf einem NT-System zu benutzen. CMD ist die NT-Entsprechung von COMMAND³.

Syntax:

```
cmd /a|/u|/q|/d|/e (ON|OFF) ||/f (ON|OFF) ||/v  
(ON|OFF) ||/c befehl||/s|/k|/y
```

Konsole: Ab Windows NT 4.0 bis Windows NT 6.1 (Windows 7)

Beispiel:

Inhalt cmd

Ausgabe (unter XP) C:\>cmd
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.
C:\>

Ausgabe (unter Vista)

C:\>cmd
Microsoft Windows [Version 6.0.6001]

3 [HTTP://DE.WIKIBOOKS.ORG/WIKI/%23COMMAND](http://de.wikibooks.org/wiki/%23COMMAND)

Copyright (c) 2006 Microsoft Corporation. Alle Rechte vorbehalten.

3.6 COLOR

Mit dem Befehl COLOR kann man die Vorder- und Hintergrundfarbe verändern. Die COLOR Werte bestehen aus zwei HEX-Werten.

0 = Black

1 = Blue

2 = Green

3 = Aqua

4 = Red

5 = Purple

6 = Darkyellow

7 = White

8 = Gray

9 = Light Blue

A = Light Green

B = Light Aqua

C = Light Red

D = Light Purple

E = Light Yellow

F = Bright White

Der Befehl

```
COLOR 2F
```

ergibt z. B. einen grünen Hintergrund mit weißer Schrift.

3.7 COMMAND

Startet einen neuen Kommandointerpreter. Dieser kann mit `exit` wieder beendet werden.

Syntax:

```
command Laufwerk:Pfad Gerät /e /l /u /P /MSG /LOW (/Y  
(/c|/k) Befehl)
```

Konsole: Ab MS-DOS bis Windows XP

Beispiel:

Inhalt `command`

Ausgabe (unter Windows 95) `C:\WINDOWS>command`

```
Microsoft(R) Windows 95  
(C)Copyright Microsoft Corp 1981-1996.  
C:\WINDOWS>
```

3.8 ECHO

Gibt einen Text aus oder schaltet die Befehlszeilen an/aus. Wenn ein Text ausgegeben wird, können dort auch VARIABLEN⁴

⁴ [HTTP://DE.WIKIBOOKS.ORG/WIKI/%23VARIABLEN](http://de.wikibooks.org/wiki/%23VARIABLEN)

angezeigt werden, wie z. B. die Variable `%ver%` (in Windows XP `%os%`).

Syntax:

`echo text|ON|OFF` oder alternativ `echo. [text]`

Interner Befehl: Ab MS-DOS bis Windows NT 5.1 (XP)

Beispiel:

Inhalt @echo off

echo Die aktuelle Datei heißt %0.

echo Die aktuelle Version Ihrer Befehls-Konsole oder -OS heißt %ver%

Ausgabe Die aktuelle Datei heißt beispiel.bat.

PAUSE

Die aktuelle Version Ihrer Befehls-Konsole oder -OS heißt Windows NT

Mit `echo.` können Sie zudem leere Zeilen ausgeben. **Beispiel:**

Inhalt @echo off

echo Jetzt gibt es 3 Leere Zeilen zu sehen!

echo.

echo.

echo.

echo So! Da waren sie.

Ausgabe Jetzt gibt es 3 Leere Zeilen zu sehen!

So! Da waren sie.

3.8.1 Benutzereingaben mit ECHO

Mit Hilfe des `echo`-Befehls können Sie auch in einem Skript Benutzereingaben simulieren, indem Sie den Pipe-Operator |

verwenden.

Beispiel: Uhrzeit anzeigen ohne Nachfrage

Das normale Verhalten des `time`-Befehls ist, die aktuelle Zeit der verwendeten Systemuhr anzuzeigen und in der nächsten Zeile die Eingabe einer neuen Uhrzeit zu erwarten. Drückt man auf Enter, bleibt die Systemzeit unverändert. Will man die Zeit nur anzeigen lassen (z. B. in einer Batchdatei vor und nach einer Befehlsfolge, um zu messen, wie lange der PC dafür braucht), lässt sich die Betätigung der Enter-Taste durch einen entsprechenden `echo`-Befehl ersetzen.

```
echo.|time
```

Über den Pipe-Mechanismus lässt sich darüber hinaus die Zeile „Geben Sie die neue Uhrzeit ein:“ unterdrücken:

```
echo.|time|find /v "neue"
```

Dabei ist die Groß-/Kleinschreibung von "neue" zu beachten oder der Schalter `/I` zu verwenden, denn `find` ist case-sensitiv! Diese Beispiele dienen allerdings nur zur Demonstration, denn die Zeitausgabe wäre auch ohne Pipes (aber erst ab Windows 2000) möglich mit: `time /t`

oder einfach: `echo %time%`

Sehr nützlich ist das `echo`-Piping auch zur Übergabe von Benutzereingaben, welche von einzelnen Befehlen abgefragt werden.

Beispiel: Überprüfung einer Festplatte

`chkdsk c: /f /r` kann (da es sich beim Laufwerk C um das Systemlaufwerk handelt) erst nach einem Systemstart ausgeführt werden. Normalerweise müsste der Benutzer deswegen den Systemstart durch Eingabe von "Y" bestätigen. Diese Aktion kann man in einem Batch so abbilden: `echo y | chkdsk c: /f /r .`

3.9 FOR

Ermöglicht die Schleifenbearbeitung.

Syntax:

```
for Variable in Satz do Befehl [Parameter]
```

Interner Befehl: Ab MS-DOS bis Windows NT 6.1 (Windows 7)

Beispiel:

Zeigt alle Dateien im Verzeichnis %temp% an. Es werden nur Dateien, keine Verzeichnisse angezeigt. Um Verzeichnisse anzuzeigen siehe Liste der FOR-Optionen unten. Der Parameter /R bewirkt, dass alle Unterverzeichnisse mit einbezogen werden (Rekursive Schleife).

ACHTUNG:

Die Variable darf nur aus einem Buchstaben bestehen! "%t" ist erlaubt, "%test" nicht! Bei der Verwendung mehrerer Befehle muss zwischen "DO" und der Klammer "(" ein Leerzeichen sein.

Falsch

```
for Variable in Satz do(
```

RICHTIG

```
for Variable in Satz do (  
    Befehl1  
    Befehl2  
)
```

Inhalt @echo off

```
for /R %temp% %%f in (*.*) do (  
    echo %%f  
)
```

REM Den Befehl könnte man auch einzeilig schreiben.

pause

Ausgabe (Alle Temp-Dateien)

Bitte beliebige Taste drücken...

Zählschleifen

Mit solchen Schleifen kann man Aktionen eine bestimmte Anzahl oft ausführen. Dazu muss man den Parameter /L angeben.

Syntax: for /L {Variable} IN (Startzahl, Schrittweite, Endzahl) DO (Aktion)

```
REM Schreibe Text 5 Mal
```

```
for /L %%N IN (1, 1, 5) DO echo Nummer %%N
```

Ausgabe:

```
C:\>for /L %%N IN (1, 1, 5) DO echo Nummer %%N
```

```
C:\>echo Nummer 1
```

```
Nummer 1
```

```
C:\>echo Nummer 2
```

```
Nummer 2
```

```
C:\>echo Nummer 3
```

```
Nummer 3
```

```
C:\>echo Nummer 4
Nummer 4
C:\>echo Nummer 5
Nummer 5
```

verschachtelte Zählschleife:

```
for /L %%N IN (1, 1, 5) DO
for /L %%N IN (1, 1, %N) DO echo Nummer %N
```

Ausgabe:

```
C:\>for /L %%N IN (1 1 5) DO (for /L %N IN (1 1 %N) DO echo Nummer %N
)
C:\>(for /L %%N IN (1 1 1) DO echo Nummer %N )
C:\>echo Nummer 1
Nummer 1
C:\>(for /L %%N IN (1 1 2) DO echo Nummer %N )
C:\>echo Nummer 1
Nummer 1
C:\>echo Nummer 2
Nummer 2
C:\>(for /L %%N IN (1 1 3) DO echo Nummer %N )
C:\>echo Nummer 1
Nummer 1
C:\>echo Nummer 2
Nummer 2
C:\>echo Nummer 3
Nummer 3
C:\>(for /L %%N IN (1 1 4) DO echo Nummer %N )
C:\>echo Nummer 1
Nummer 1
C:\>echo Nummer 2
```

```
Nummer 2
C:\>echo Nummer 3
Nummer 3
C:\>echo Nummer 4
Nummer 4
C:\>(for /L %%N IN (1 1 5) DO echo Nummer %N )
C:\>echo Nummer 1
Nummer 1
C:\>echo Nummer 2
Nummer 2
C:\>echo Nummer 3
Nummer 3
C:\>echo Nummer 4
Nummer 4
C:\>echo Nummer 5
Nummer 5
```

Weitere Möglichkeiten der FOR-Schleife:

syntax-FOR-Files

```
FOR %%parameter IN (set) DO command
```

syntax-FOR-Files-Rooted at Path

```
FOR /R [[drive:]path] %%parameter IN (set) DO command
```

syntax-FOR-Folders

```
FOR /D %%parameter IN (folder_set) DO command
```

syntax-FOR-List of numbers

```
FOR /L %%parameter IN (start,step,end) DO command
```

syntax-FOR-File contents

```
FOR /F ["options"] %%parameter IN (filename) DO command
```

```
FOR /F ["options"] %%parameter IN ("Text string to process") DO
command
```

syntax-FOR-Command Results

```
FOR /F ["options"] %%parameter IN ('command to process') DO
command
```

Beispiel: Sucht im Ordner C:\Windows\Temp rekursiv nach Dateien mit dem Namen //temp.dat// und gibt die Liste aus. Die Option "token=*" ist notwendig, damit die Ausgabe zeilenweise gelesen wird und auch Pfade mit enthaltenem Leerzeichen ausgegeben werden können.

```
for /F "tokens=*" %%f in ('dir /S /b C:\Windows\Temp\temp.dat') do (
    echo "%%f".
)
```

Zählvariablen in Zeichenketten einbetten

Um die Zählvariable %%f (%%f auf der Kommandozeile) in einer Zeichenfolge zu verwenden, wird einfach die Variable in dem String eingebettet

```
FOR %%f IN (A B C D E) DO (
```

```
    echo mitten%%fdrinnen
)
```

3.10 GOTO

Mit dem Batchbefehl `goto` wird eine Sprungmarke : (s.o.) angesprungen.

Syntax

```
goto NAMEDERSPRUNGMARKE
```

Beispiel

Siehe unter : (DOPPELPUNKT)⁵.

3.11 IF

Der `IF` Befehl ermöglicht eine einfache Verzweigung und wird oft zusammen mit dem `GOTO`⁶ Befehl eingesetzt. `IF` ermöglicht hierbei sowohl die Prüfung auf eine Gleichheit als auch auf das Vorhandensein von Dateien.

Beispiel 1:

```
IF exist c:\temp\my.log echo.>c:\temp\my.log
echo.Log Datei erstellt>>c:\temp\my.log
```

5 [HTTP://DE.WIKIBOOKS.ORG/WIKI/%23%3A%20%28DOPPELPUNKT%29](http://de.wikibooks.org/wiki/%23%3A%20%28DOPPELPUNKT%29)

6 [HTTP://DE.WIKIBOOKS.ORG/WIKI/%23GOTO](http://de.wikibooks.org/wiki/%23GOTO)

Beispiel 1 prüft ob eine Logdatei vorhanden ist und erstellt ggf. eine Neue.

Beispiel 2:

```
IF "%COMPUTERNAME%" == "Bastie" GOTO :WAHR
REM hier landet man wenn der if-Ausdruck falsch ist
    GOTO WEITER
:WAHR
REM hier landet man wenn der if-Ausdruck wahr ist
    echo Willkommen Zuhause
    REM Jetzt wird der if Zweig verlassen
    GOTO WEITER

:WEITER
echo.Have a nice Day!
```

Beispiel 3:

```
IF "%COMPUTERNAME%" == "Bastie" (
    echo Willkommen zu Hause!
) ELSE (
    echo Du bist auf Computer: %COMPUTERNAME%
)
echo. Schönen Tag noch!
```

Beachten Sie, bei der Prüfung von Umgebungsvariablen niemals

```
IF %Umgebungsvariable% == Prüfwert ...
```

zu schreiben, wenn die Umgebungsvariable nicht gesetzt ist; Sie erhalten sonst einen Syntaxfehler. Der Parameter /i unterbindet eine Differenzierung der Groß-/Kleinbuchstaben.

ACHTUNG:

Bei der Verwendung mehrerer Befehle muss zwischen *Bedingung* und der Klammer "(" ein Leerzeichen sein.

Falsch

```
IF Bedingung (
```

Richtig

```
IF Bedingung (
    Befehl1
    Befehl2
)
```

3.11.1 Syntax Vergleiche

IF <NOT>Variable1==Variable2

NOT Der Befehl wird nur ausgeführt, wenn die Bedingung NICHT Wahr ist. Optional.

== ist gleich

EQU ist gleich

NEQ nicht gleich

LSS kleiner als

LEQ kleiner als oder gleich

GTR größer als

GEQ größer als oder gleich

3.12 PAUSE

Unterbricht die Abarbeitung der Batchdatei und wartet auf einen Tastendruck.

Syntax

pause

Interner Befehl: Ab MS-DOS bis Windows NT 5.1 (XP)

Beispiel 1:

Inhalt @echo off
echo Willkommen zur Batchdatei %0 !!!
echo.
echo Die Batchdatei wird auf einer neuen Seite fortgeführt.
pause
cls
echo Hier fängt meine Batchdatei an...
pause

Ausgabe Willkommen zur Batchdatei beispiel.bat !!!
Die Batchdatei wird auf einer neuen Seite fortgeführt.
Bitte beliebige Taste drücken...

(Neue Seite) Hier fängt meine Batchdatei an...
Bitte beliebige Taste drücken...

Beispiel 2:

Inhalt @echo off
echo Diese Zeile wird angezeigt anstelle von "Bitte beliebige Taste
drücken..."
pause > NUL

Ausgabe Diese Zeile wird angezeigt anstelle von "Bitte beliebige Taste
drücken..."

Um eine Pause für eine bestimmte Zeitdauer vorzugeben, kann der PING-BEFEHL⁷ „missbraucht“ werden. Das folgende Beispiel

⁷ [HTTP://DE.WIKIBOOKS.ORG/WIKI/BATCH-{}PROGRAMMIERUNG%3A%20NETZWERKUMGEBUNG%23PING](http://de.wikibooks.org/wiki/Batch-{}programmierung%3A%20Netzwerkumgebung%23ping)

erzeugt eine Pause von etwa 10 Sekunden (nämlich 11 minus 1); durch die Ausgabeumleitung `>nul` wird jegliche Meldung unterdrückt. `ping -n 11 localhost >nul`

3.13 REM

REM leitet einen Kommentar ein. Die Zeile wird ignoriert, beachten Sie jedoch, dass REM von einem Leerzeichen / Tabulator gefolgt werden muss. Alternativ dazu werden häufig auch Sprungmarken eingesetzt, da hier nur ein statt vier Zeichen verwendet werden muss. Häufig wird jedoch die Sprungmarke zur besseren Übersicht doppelt hintereinander geschrieben : :

Beispiel:

```
REM kill iexplore.exe
:kill iexplore.exe
::kill iexplore.exe
```

3.14 START

Startet ein Programm.

Syntax:

```
START ["Titel"] [/D <Pfad>] [/I] [/MIN] [/MAX]
[/SEPARATE | /SHARED] [/LOW | /NORMAL | /HIGH
| /REALTIME] [/WAIT] [/B] [Befehl/Programm]
[Parameter]
```

Optionen:

- "Titel" Der Titel des neuen Fensters.

- /D <Pfad> Startverzeichnis
- /I Die neue Umgebung soll die dem CMD.EXE beim Aufruf übergebene sein und nicht die aktuelle Umgebung.
- /MIN Startet das Fenster minimiert.
- /MAX Startet das Fenster maximiert.
- /SEPARATE Startet 16-Bit-Windows-Programm in separatem Speicherbereich.
- /SHARED Startet 16-Bit-Windows-Programm in gemeinsamen Speicherbereich.
- /LOW Startet Anwendung in IDLE-Prioritätsklasse.
- /NORMAL Startet Anwendung in der NORMAL-Prioritätsklasse.
- /HIGH Startet Anwendung in der HIGH-Prioritätsklasse.
- /REALTIME Startet Anwendung in der REALTIME-Prioritätsklasse.
- /WAIT Startet die Anwendung und wartet auf das Ende.
- /B Startet die Anwendung ohne ein neues Fenster zu öffnen. Die Anwendung ignoriert STRG+C. Wenn die Anwendung nicht selbständig STRG+C überprüft, ist STRG+UNTBR die einzige Möglichkeit, um die Anwendung abzubrechen.
- /? Gibt die Hilfe aus.

Hier eine Liste der nützlichen (System)Programme, die man so ausführen kann:

regedit.exe = neuer Registrierungseditor

(regedt32.exe = älterer Registrierungseditor)

explorer.exe = Windows Ordner Explorer

taskmgr.exe = Windows Taskmanager

taskeng.exe = Aufgabenplanungsmodul

calc.exe = Taschenrechner

mshta.exe = Scripthost für HTA (HTML) Scripting

W/CScript.exe = Beides Bestandteil des Windows Scripthostes für JS und VBS

iexplore.exe = Microsoft Windows Internet Explorer, Standardbrowser für Windows

firefox.exe = Mozilla Firefox, Internetbrowser

dialer.exe = Windows-Hilfeprogramm für DFÜ
Einwahlverbindungen

Notepad.exe = Standard-Textbearbeitungsprogramm von
Microsoft

cmd.exe = Microsoft Windows Befehlsprozessor für Batch

3.15 Variablen

Es gibt zweierlei Arten von Variablen: Scriptvariablen und Systemvariablen. Die Scriptvariablen werden innerhalb von Schleifen eingesetzt, beginnen mit % und einem frei wählbaren Zeichen,

oder einer Zeichenfolge. Systemvariablen müssen von % begonnen und von % abgeschlossen werden. Sie sind fest definiert. Die Variable %0% hat stets den Namen der aktuellen Datei. Die Variable %ver% hat immer den Namen der aktuellen Betriebssystemversion. Unter Windows XP muss %os% statt %ver% verwendet werden.

Gibt man unter Windows Vista und 7 den Befehl `echo %os%` ein, so gibt der Computer die Version des Systems aus, auf dem das System basiert. Unter Windows Vista und Windows 7 gibt es einen Befehl `namen ver`, mit dem man die Version direkt angezeigt bekommt. Es gibt zwar keine direkte Variable bei Vista und Windows 7, die das Betriebssystem definiert, jedoch kann man das mit folgendem Trick umgehen:

```
@echo off           ' Autobefehlsanzeige ausgeschaltet
ver > TMP.dat       ' speichert die Ausgabe des Befehls in der
Datei TMP.dat
Set /p ver= < TMP.dat ' definiert die Variable des Betriebssystems.
(NUR TEMPORÄR !)
echo %ver%          ' gibt nun die definierte Variable aus.
pause>nul           ' pause
```

Nützliche Stringoperation für Variablen:

Teilstring

Syntax

```
var:~n,m
```

Ergibt den Teilstring von `var`, beginnend mit dem `n`-ten Zeichen (von links) und einer Länge von `m` Zeichen. Gezählt wird ab 0, d.h. das erste Zeichen hat die Position 0 und nicht 1. Werden negative Werte verwendet, so wird vom Ende des Strings /von rechts nach links) gezählt.

Beispiel:

Inhalt:

```
set str=myfile.bat
set name=%str:~0,6%
echo %name%
```

In diesem Beispiel wird der Teilstring von str vom ersten Zeichen an mit einer Länge von 6 Zeichen ausgegeben.

Ausgabe:

```
myfile
```

Stringsstitution

Syntax

```
var:str1=str2
```

Mithilfe des Syntax `var:str1=str2` kann die Zeichenkette `str1` des Inhalts der Variable `var` durch `str2` ersetzt werden.

Beispiel:

Inhalt:

```
set str="mycommand /p /m file"
echo %str%
set str=%str:/p /m=/t %
echo %str%
```

In diesem Beispiel werden die Optionen `/p /m` des Befehls `mycommand` durch die Option `/t` ersetzt.

Ausgabe:

```
"mycommand /p /m file"
"mycommand /t file"
```

Benutzereingaben in Variablen speichern

Um eine Benutzereingabe in eine Variable speichern zu können wird die Option /p benötigt

Beispiel:

```
set /p EINGABE=Variable Eingeben:  
echo %EINGABE%
```

Mit Variablen Rechnen

Rechnungen mit Variablen sind erst mit der Option /a möglich

Beispiel:

```
set WERT1=2  
set WERT2=21  
set /a WERT1 = WERT1 * WERT2  
echo %WERT1%
```

3.16 TASKKILL

Beendet Prozesse direkt über den Taskmanager. (Nur unter Windows XP Professional, Windows Vista und Windows 7)

Syntax

```
taskkill /IM prozessname.dateiendung
```

Beispiel:

Inhalt:

```
@echo off  
echo Der Internetexplorer wird gestartet und in 5 Sekunden geschlossen  
start iexplore http://de.wikibooks.org
```

```
@ping localhost -n 5 >NUL  
taskkill /IM iexplore.exe  
pause
```

In diesem Beispiel wird die Anwendung Internetexplorer gestartet und 5 Sekunden nach dem Start wieder geschlossen. Das Kommandozeilenfenster bleibt offen, bis der Benutzer eine beliebige Taste drückt.

4 Batch Operatoren

4.1 &

Befehlsverkettung: mehrere Batch-Befehle in einer Zeile können hintereinander ausgeführt werden

Syntax

```
befehl1 & befehl2
```

Stapelanweisung: Ab ? bis Windows 7

Beispiel:

Inhalt `echo Hallo! & echo und Tschüss!`

Ausgabe `Hallo!`

`und Tschüss!`

4.2 |

Befehlsverkettung mittels "pipe": der zweite Befehl bekommt die Ausgabe des ersten Befehls als Eingabe

Syntax

```
befehl1 | befehl2
```

Stapelanweisung: Ab DOS (nur für more als zweiter Befehl); allgemein verwendbar ab WindowsNT bis Windows 7

Beispiel:

Inhalt `type test.txt | more`

Ausgabe Zeile 1

Zeile 2

...

Zeile 24

-- More --

(nach Drücken einer beliebigen Taste werden die nächsten Zeilen angezeigt)

4.3 &&

bedingte Befehlsverkettung: der zweite Befehl wird nur ausgeführt, wenn der erste Befehl erfolgreich war

Syntax

`befehl1 && befehl2`

Stapelanweisung: Ab ? bis Windows 7

Beispiel:

Inhalt `copy test.txt test2.txt && echo Kopieren erfolgreich!`

Ausgabe `Kopieren erfolgreich!`

(falls test.txt existiert und nach test2.txt kopiert werden konnte)

4.4 ||

bedingte Befehlsverkettung: der zweite Befehl wird nur ausgeführt, wenn der erste Befehl fehlschlug

Syntax

```
befehl1 || befehl2
```

Stapelanweisung: Ab ? bis Windows 7

Beispiel:

Inhalt `copy test.txt test2.txt || echo Fehler beim Kopieren!`

Ausgabe Fehler beim Kopieren!

(falls test.txt nicht existiert oder test2.txt nicht geschrieben bzw. überschrieben werden konnte)



4.5 <

Umleitung der Eingabe

Syntax

```
befehl < file
```

Beispiel:

Inhalt `echo Hallo! > tmp.txt`

```
set /P v= < tmp.txt
```

```
echo %v%
```

Ausgabe Hallo!

(set /P v= würde eine Zeile vom Bediener erwarten; durch die Umleitung wird diese Zeile von der Datei tmp.txt gelesen, die zuvor mittels Umleitung der Ausgabe (siehe unten) befüllt wurde.)

4.6 >

Umleitung der (Standard-)Ausgabe zu einem anderen Ziel. Wenn das Ziel eine Datei ist, wird diese neu angelegt (falls die Datei schon existiert, wird sie zuvor gelöscht) Es kann aber auch nach NUL(Die Ausgabe verschwindet) oder CON(Bildschirm) umgeleitet werden

Syntax

```
befehl > file
```

oder auch mit vorrangestellter Umleitung (praktisch wenn bei der Ausgabe einzelne " vorkommen)

```
> file befehl
```

Beispiel:

```
Inhalt echo Hallo! > tmp.txt  
type tmp.txt
```

```
Ausgabe Hallo!
```

Neben der Standard-Ausgabe (**stdout**) gibt es noch eine Fehlerausgabe (**stderr**). Normalerweise landen die Standard-Ausgabe und die Fehler-Ausgabe am Bildschirm, wodurch der Unterschied nicht auffällt. Wenn die Standard-Ausgabe auf eine Datei umgelenkt wird, landen Fehlerausgaben immer noch auf dem Bildschirm, was oft erwünscht ist. Falls nicht, kann auch die Fehler-Ausgabe in eine Datei umgeleitet werden, und zwar mit 2> .

Beispiel: `dir >stdout.txt 2>stderr.txt` Schreibt den Output des `dir` Befehles in die Datei *stdout.txt*, da kein Fehler aufgetreten ist.

`dir DieseDateiExistiertNicht.txt >stdout.txt 2>stderr.txt`

Schreibt die Fehlermeldung des `dir` Befehles in die Datei *stderr.txt*, da die Datei *DieseDateiExistiertNicht.txt* nicht existiert und somit ein Fehler auftritt.

Wenn die Fehlerausgabe in die gleiche Datei wie die Standardausgabe umgeleitet werden soll, hängt man noch `2>&1` an.

Beispiel:

`type EineDatei.txt > UmgeleiteteAusgabe.txt 2>&1`

Der Inhalt der Datei *EineDatei.txt* wird in die Datei *UmgeleiteteAusgabe.txt* kopiert (wenn *EineDatei.txt* existiert).

`type EineDateiDieNichtExistiert.txt > UmgeleiteteAusgabe.txt 2>&1`

Wenn die Datei *EineDateiDieNichtExistiert.txt* nicht existiert, dann wird die Fehlermeldung nicht im Dos-Fenster (**stdout**) ausgegeben, sondern die Ausgabe wird auch in die Datei *UmgeleiteteAusgabe.txt* geschrieben.

4.7 >>

Umleitung der (Standard-)Ausgabe mit Anhängen des Textes (falls die Datei schon existiert; sonst wird die Datei wie bei `>` (siehe oben) angelegt)

Syntax

`befehl >> file`

Stapelanweisung: Ab ? bis Windows 7

Beispiel:

Inhalt echo Hallo! > tmp.txt
echo Haalooo!! >> tmp.txt
type tmp.txt

Ausgabe Hallo!
Haalooo!!

Hier gilt analoges für die Fehler-Ausgabe (siehe Operator '>' oben).

5 Erweiterungen unter Windows NT

Unter Windows NT, Windows 2000 und Windows XP stehen einige weitere Befehle zur Verfügung, zum Teil sind dort auch erweiterte Optionen für Standard-Befehle enthalten.

5.1 Batchparameter

Die Parameter lassen sich jetzt folgendermaßen erweitern (für 1 setze man den jeweiligen Parameter ein):

```
%~l  Anführungszeichen (") werden entfernt
%~f1 vollständige Pfadbezeichnung
%~d1 Laufwerksbuchstabe
%~p1 Pfad (ohne Laufwerksbuchstabe)
%~n1 Dateiname
%~x1 Dateinamenserweiterung
%~s1 Pfad nur mit kurzen Verzeichnis/Dateinamen
([http://de.wikipedia.org/wiki/8.3 8.3-Konvention])
%~a1 Dateiattribute
%~t1 Datums- und Uhrzeitangaben der Datei (Geändert am)
%~z1 Größe der Datei
```

Die Angaben können auch kombiniert werden, z.B. %~dp0 ist Laufwerksbuchstaben + Pfad der aktuellen Batchdatei.

5.2 Stringmanipulation

Mit folgender Syntax lassen sich Teile aus einer Zeichenkette extrahieren:

```
set str=Hallo
set str=%str:~1%
echo.%str%
```

Die angegebene positive Zahl gibt an, wie viele Zeichen links übergangen werden sollen. Dieser Ausdruck liefert daher den Teilstring "allo" von "Hallo".

```
set str=Hallo
set str=%str:~-4%
echo.%str%
```

Wird eine negative Zahl eingesetzt, so gibt ihr Betrag an, wieviele Zeichen von rechts genommen werden sollen. Dieser Ausdruck liefert daher ebenfalls "allo".

Über die Angabe einer zweiten Zahl kann definiert werden, was mit dem Rest nach Auswertung der ersten Zahl entsprechend obiger Regeln geschehen soll. Während die erste Zahl die Zeichenkette links beschneidet, beschneidet die zweite Zahl sie rechts und zwar nach folgenden Regeln:

```
set str=Hallo
set str=%str:~1,2%
echo.%str%
```

Eine positive zweite Zahl gibt an wie viele Zeichen ab links vom Rest genommen werden sollen. Dieser Ausdruck liefert daher "al".

```
set str=Hallo
set str=%str:~1,-2%
echo.%str%
```

Wird eine negative zweite Zahl eingesetzt, so gibt ihr Betrag an, wie viele Zeichen vom Rest ab rechts übergangen werden sollen. Dieser Ausdruck liefert daher ebenfalls "al".

Möchte man beispielsweise nur einen Teil der Zeichenkette ab links, so ist nach diesen Regeln als erste Zahl eine Null anzugeben:

```
set str=Hallo
set str=%str:~0,2%
echo.%str%
```

Dieser Ausdruck liefert daher "Ha".

MEHR ZU STRINGMANIPULATION¹

5.3 Umgebungsvariablen

Unter DOS und Windows 9x/ME speichern alle Anwendungen und Skripte ihre Umgebungsvariablen in der gleichen Umgebung. Unter Windows NT-basierten Betriebssystemen ist dies etwas anders:

Zunächst gibt es einen systemweiten Bereich. Dieser enthält Variablen, die für alle Benutzer und alle Anwendungen zugänglich sind. Zusätzlich hat jeder Benutzer seinen eigenen Variablenraum. Jede Anwendung, die nun vom System gestartet wird, erhält einen eigenen Variablenraum, der als Kopie aus dem der aufrufenden

¹ [HTTP://WWW.DOSTIPS.COM/DtTipsStringManipulation.PHP#_Toc135152735](http://www.dostips.com/DtTipsStringManipulation.php#_Toc135152735)

Anwendung erzeugt wird. Ändert das Programm etwas (Wertänderung, neue Variable), wirkt sich dies nicht auf parallel gestartete Anwendungen oder das System aus. Lediglich Prozesse, die von der Anwendung gestartet werden erhalten eine Kopie des Anwendungsvariablenraumes. Dadurch ist es nicht möglich, dass Anwendungen Umgebungsvariablen setzen, die dann anschließend in einer Batchdatei verwertbar sind.

Beispiel:

```
@echo off
set a=2
echo %a%
command /c: set a=1
echo %a%
```

Unter Dos würde die Ausgabe lauten:

```
2
1
```

Unter Windows NT/2000/XP:

```
2
2
```

Das heißt nicht, dass unter Windows NT und Nachfolgern der Befehl `set a=1` keinen Effekt hätte, sondern dieser wird in einer geschützten Umgebung ausgeführt, die dann außerhalb dieser Umgebung keinen Effekt mehr hat. Für die aufgerufene Variante von `command` ist `a` nach dem `set`-Befehl sehr wohl gleich 1.

NOCH EINE ÄNDERUNG VARIABLEN BETREFFEND.²

5.4 Die Kommandozeile lernt Rechnen

Seit Windows NT 4 kann man mit dem `set` Befehl auch Rechnen.

Ein Beispiel: `set /a 1+2`

3

Man kann das Ergebnis auch in Variablen schreiben: `set /a n=5+5`

10

`echo %n%`

10

- CMD nimmt folgendes entgegen (Auszug aus der Hilfe):

<code>()</code>	- Gruppierung
<code>! ~ -</code>	- unäre Operatoren
<code>* / %</code>	- arithmetische Operatoren (% bedeutet Modulo)
<code>+ -</code>	- arithmetische Operatoren
<code><< >></code>	- logische Verschiebung
<code>&</code>	- bitweise UND
<code>^</code>	- bitweise exklusives ODER
<code> </code>	- bitweise ODER
<code>= *= /= %= += -=</code>	- Zuordnung

- Mit Klammern:

`set /a (1+1)*(4-3)`

2

² [HTTP://DE.WIKIBOOKS.ORG/WIKI/BATCH-{}PROGRAMMIERUNG:_-_PROGRAMMIERUNGS_HILFEN#PROBLEME_MIT_VARIABLEN](http://de.wikibooks.org/wiki/Batch-{}_Programmierung:_-_Programmierungs_Hilfen#Probleme_Mit_Variablen)

5.5 Benutzereingaben

Seit Windows 2000 kann man mit dem `set` Befehl auch Eingaben abfragen:

Syntax: `set /p {variablenname}={Angezeigter Text}`

5.5.1 Beispiel

```
@echo off
set /p var=Eingabe:
echo Sie haben %var% eingegeben.
pause
```

Ausgabe: Eingabe: Test
Sie haben Test eingegeben.
{Pause}

Prüfung ob der Benutzer Text eingegeben hat: `@echo off`

```
set /p var=Eingabe:
if not defined var (
    echo Bitte geben Sie etwas ein!
    set var=niehts
    REM Könnte auch was ganz anderes sein
)
echo Sie haben "%var%" eingegeben.
pause
```

Ausgabe: Eingabe: {leer}
Bitte geben Sie etwas ein!
Sie haben "niehts" eingegeben.

```
{pause}
```

5.6 Ausgaben in eine Variable umleiten

Eine Möglichkeit der Umleitung einer Ausgabe besteht über den `set /p` Befehl:

Syntax: `ECHO {Befehl}> {Dateiname}`
`SET /P {Variablenname}=<{Dateiname}`

5.6.1 Beispiel

```
@ECHO OFF
ECHO %date% > _.txt
SET /P tmp=<_.txt
ECHO Das Datum heute ist: %tmp%
```

- siehe auch [NET-BEFEHLE](#)³

³ [HTTP://DE.WIKIBOOKS.ORG/WIKI/BATCH-{}PROGRAMMIERUNG%3A%20NETZWERKUMGEBUNG%23NET](http://de.wikibooks.org/wiki/Batch-{}_Programmierung%3A%20Netzwerkumgebung%23net)

6 Programmierungshilfen

6.0.2 Ändern des Editors zum Bearbeiten von Batchdateien

Wenn man im Windows Explorer mit der Rechten-Maus-Taste (RMT) auf eine *.bat klickt, so werden einem die Befehle *Öffnen* und *Bearbeiten* angeboten.

Öffnen: führt die Batchdatei aus. Mit dem Befehl

Bearbeiten: wird die Batchdatei in den Texteditor *notepad.exe* zum Bearbeiten geöffnet.

Auch wenn Notpad zum Bearbeiten von Batchdateien ausreicht, so möchte man häufig doch die Batchdateien mit einem anderen, komfortableren Editor bearbeiten, der z.B. Syntaxhervorhebung (Syntaxhighlighting) beherrscht.

Um einen anderen Editor (z.B. SYN¹) zu verwenden, muss man in der Registry an der Stelle:

```
HKEY_CLASSES_ROOT\batfile\shell\edit\command
```

1 [HTTP://SYN.SOURCEFORGE.NET/](http://syn.sourceforge.net/)

den Standard Wert

```
(Standard) = %SystemRoot%\System32\notepad.exe %1
```

auf den Startbefehl des entsprechenden Editors ändern. Z.B.

```
HKEY_CLASSES_ROOT\batfile\shell\edit\command | (Standard) =  
c:\Programme\Editor\syn\syn.exe %1
```

6.0.3 Datum und Uhrzeit anzeigen

Für Log-Dateien ist es wichtig, dass man die Logeinträge mit Datum und Uhrzeit versehen kann:

```
echo %date:~0% - %time:~0,8% Uhr
```

Ergebnis: **11.09.2010 - 15:59:53 Uhr**

Hierbei steht ":~0,8" für die Angabe der Stellen. Mit "0,8" wird angegeben, dass die Ausgabe der Zeit bei Position 0 beginnen soll und insgesamt 8 Stellen beinhalten soll. Die maximale Stellenanzahl ist 11 (0,11).

Beim Datum ist diese Angabe hier nicht nötig, da dieses standardmäßig im dd.mm.yyyy-Format ausgegeben wird. Wer aber nur das Jahr haben will, kann "%date:~-4%" eingeben und erhält damit die letzten 4 Zeichen. Für ein sortiergerechtes Datum in der Umgebungsvariablen sortdate sorgt z.B.

```
set SORTDATE=%date:~-4%-%date:~3,2%-%date:~0,2%  
echo %SORTDATE%
```

Wert der Umgebungsvariable: **2009-04-20**

Beachte:

In einer Batchdatei kann die Verwendung von %DATE% und insbesondere von %TIME% dazuführen, das die ausgegebene Uhrzeit sich nicht aktualisiert.

Hierzu folgendes Beispiel:

```
@echo off
echo ## Die aktuelle Zeit ist: %TIME%
echo ## bitte ca. 5 Sec. warten ...
ping -n 5 localhost>NUL
echo ## jetzt sollten ca. 5 Sec. vergangen sein, TIME liefert %TIME%,
    das ist noch OK
echo ## doch in der FOR Schleife wird bereits die alte Zeit verwendet.
for /L %%N IN (0, 1, 3) DO (
    echo %time%
    pause
)
echo ## und dies bleibt fuer jede Ausgabe innerhalb der FOR-Schleife
so.
echo.
echo ## Auch in z.B. IF-Schleifen ist das so.
if TRUE==TRUE (
    echo 1. Zeit in der If Schleife: %TIME%
    echo Warte ca. 5 Sec.
    ping -n 5 localhost >NUL
    echo 2. Zeit in der If Schleife: %TIME%
    echo Warte nochmals ca. 5 Sec.
    ping -n 5 localhost >NUL
    echo 3. Zeit in der If Schleife: %TIME%
)
echo ## Dabei ist es bereits: %TIME%
pause
```

Damit %DATE% und %TIME% die richtigen Werte ausgeben, muss unbedingt die verzögerte Erweiterung von Umgebungsvari-

ablen mit dem Befehl `SETLOCAL2 ENABLEEXTENSIONS` aktiviert werden.

Hier das korrekte Beispiel:

```
@echo off
SetLocal EnableDelayedExpansion
echo ## Die aktuelle Zeit ist: %time:~0,8%
echo ## bitte ca. 5 Sec. warten ...
ping -n 5 localhost>NUL
echo ## jetzt sollten ca. 5 Sec. vergangen sein, TIME liefert %TIME%,
    das ist OK
echo ## Jetzt gibt auch die FOR Schleife die korrekte Zeit aus.
for /L %%N IN (0, 1, 3) DO (
    echo !TIME!
    pause
)
echo.
echo ## Auch in z.B. IF-Schleifen ist es jetzt richtig.
if TRUE==TRUE (
    echo 1. Zeit in der If Schleife: %TIME%
    echo Warte ca. 5 Sec.
    ping -n 5 localhost >NUL
    echo 2. Zeit in der If Schleife: %TIME%
    echo Warte nochmals ca. 5 Sec.
    ping -n 5 localhost >NUL
    echo 3. Zeit in der If Schleife: %TIME%
)
echo ## Es ist jetzt: %TIME%
EndLocal
pause
```

2 [HTTP://DE.WIKIBOOKS.ORG/WIKI/BATCH-{}PROGRAMMIERUNG%3A%20WICHTIGE_DOS-{}KOMMANDOS%23SETLOCAL](http://de.wikibooks.org/wiki/Batch-{}PROGRAMMIERUNG%3A%20WICHTIGE_DOS-{}KOMMANDOS%23SETLOCAL)

6.0.4 Ausgaben besser anzeigen

Wenn man nicht die Ausgabe von Befehlen per **@echo off** "Ausblendet" kann man am besten das Prompt ändern, sodass man besser erkennen kann, was passiert:

```
@prompt -&G
```

Der Prompt ist dann ->

6.0.5 Unterrouninen und Unterprogramme

Unterrouninen kann man mittels **goto** oder **call** und Unterprogramme mit Hilfe von **call** realisieren.

```
call:unterroutine Hallo
echo Fertig!
goto:eof
:unterroutine
    echo Übergebener Parameter an Unterroutine: %1
goto:eof
```

Beachten Sie, dass beim Aufruf von Unterrouninen per **call** Sie Probleme mit Filehandles bekommen können. Dies liegt darin begründet, dass ein Aufruf per **call** als Aufruf eines Unterprogramms interpretiert wird, während es sich bei **goto** stets um Unterrounines handelt.

Anmerkung: **goto:eof** ist eine Spezialmarke mit der Sie stets zum Ende Ihres Skriptes (bzw. Ihrer Unterrounine) springen

6.0.6 Benutzereingaben mittels "set /P"

```
@echo off
    set /P w= [i]nstallieren / [d]eInstallieren?
    REM die option /I beim if bewirkt, dass nicht
    REM zwischen Gross und Kleinschreibung
    REM unterschieden wird.
    if /I "%w%"=="i" goto Install
    if /I "%w%"=="d" goto Deinstall
    echo Fehler: [%w%]
goto ende
:Install
    echo "installieren" ausgewählt
goto ende
:Deinstall
    echo "deInstallieren" ausgewählt
goto ende
:ende
    echo.
    pause
```

6.0.7 stdout in Umgebungsvariable speichern

Falls man den **stdout** in einer Umgebungsvariable speichern möchte, muss man das komplizierter umsetzen. Es gibt zwei verschiedene Möglichkeiten dies anzugehen. `befehl | set /P variable=` funktioniert nämlich nicht. Stattdessen braucht man:

```
BEFEHL > temp.txt
set /p BefehlOutput= < temp.txt
```

```
del temp.txt
```

Oder:

```
FOR /F %%i IN ('BEFEHL') DO set BefehlOutput=%%i
```

Oder mit "usebackq"-Option:

```
FOR /F "usebackq" %%i IN (`BEFEHL`) DO set BefehlOutput=%%i
```

Die Zeichenkette zwischen den einfachen Anführungszeichen wird dabei als Befehlszeile betrachtet und von einer untergeordneten CMD.EXE ausgeführt. %BefehlOutput% kann nun beliebig gebraucht werden.

Beispiel:

Code:

```
@echo off
FOR /F %%i IN ('CD') DO set verzeichnis=%%i
echo %verzeichnis%
```

Ausgabe:

```
C:\Programme\Batch
```

Vorsicht ist geboten, bei Befehlen, welche mehrzeilige Ausgaben produzieren und bei solchen, welche in ihrer Ausgabe auch Leerzeichen enthalten können. Da das Standardtrennzeichen ein Blank ist muss man, wenn man nicht will, dass die Variable nur bis zum Blank gefüllt wird, das Standardtrennzeichen verändern. `FOR /F "delims=" %%i IN ('CD') DO set verzeichnis=%%i` entfernt jede Art von Trennzeichen. Bei Be-

fehlen, welche mehrzeilige Ausgaben zur Folge haben, bleibt jeweils die letzte Zeile in der Variablen erhalten.

6.0.8 Dateien und Verzeichnisse auflisten

Hier ist ein Beispiel, in dem alle Dateien, auf welche die Filterbedingung zutrifft, aufgelistet werden. Ausserdem werden die Dateianzahl und die Dateigrößen zusammenaddiert.

```
@echo off
set Filter=*. *
set /A DateiAnzahl=0
set bytes=0
for /R %pfad% %%f in (%Filter%) do (
    set /A DateiAnzahl += 1
    echo %%f - %%~zfBytes
    set /A bytes=bytes+%%~zf
)
echo.
echo %~dp0%Filter%
echo Es sind %DateiAnzahl% Dateien vorhanden.
echo Alle Dateien zusammen: %bytes%Bytes
set /A kbytes=bytes/1024
echo umgerechnet sind das %kbytes% KBytes
echo.
pause
```

6.0.9 Pause

Oft ist es hilfreich, dass ein nach dem Beenden des Batch-Programms das Eingabeaufforderungsfenster offen bleibt. So

kann man Ausgaben nachlesen oder evtl. aufgetretene Fehler entdecken. Nun könnte man einfach am Ende eine **pause** einfügen. Dabei kann der User einfach das Fenster schließen oder ENTER drücken. Man kann aber auch einfach eine zeitliche Pause mit **ping** realisieren:

```
@echo off
echo Ich schließe gleich.
@ping localhost -n 2 >NUL
```

Dabei kann man die Zeit mit dem Parameter `-n` variieren.

Bei installiertem RESOURCEKIT³ steht der Befehl "sleep" zur Verfügung welcher dieselbe Funktionalität (zeitliche Pause) bietet.

6.0.10 Minimiert ausführen

Hin und wieder ist es sinnvoll, dass die Batchdatei minimiert ausgeführt wird (z. B. eine Login-Batch-Datei). Es ist möglich, dass man die Batchdatei normal startet und sie sich selber minimiert ausführt. Der Nachteil ist allerdings, dass sich kurzzeitig ein Eingabeaufforderungs-Fenster öffnet.

```
@echo off
if not "%1"==" " goto %1
start /MIN cmd.exe /C "%~nx0 begin"
```

3 `HTTP://WWW.GOOGLE.DE/URL?SA=T&SOURCE=WEB&CT=RES&CD=1&VED=0CAGQFJAA&URL=HTTP%3A%2F%2FWWW.MICROSOFT.COM%2FDOWNLOADS%2FDETAILS.ASPX%3FFAMILYID%3D9D467A69-{}57FF-{}4AE7-{}96EE-{}B18C4790CFFD&RCT=J&Q=MS+RESOURCE+KIT&EI=SETGS8H6FYFM-{}QAI2TRT&USG=AFQJCNUL30WZLC_L61SAIBSG99A7LCCRG|MS`

```
goto:eof
:begin
    echo Hallo, ich laufe minimiert!
    pause
goto:eof
```

Noch eine Konstruktion ganz ohne Labels, nach DIESEM NEWSGROUP-BEITRAG⁴

```
@set !=||(set !=1&start "%~dpnx0" /min cmd /c %0 %*&set !=&goto :eof)
```

Anmerkung: Sollte die Command-Processor-Option "DelayedExpansion" in der Registry aktiviert sein (siehe unter Hilfe "cmd /?"), lässt sich ein "!" als Variablenname **nicht** verwenden. In diesem Fall -bzw. sinnvollerweise immer- den Variablennamen ändern auf x oder y oder # oder @....

Beispiel:

```
@set #=||(set #=1&start "%~dpnx0" /min cmd /c %0 %*&set #=&goto :eof)
```

6.0.11 Mittels start /LOW die Priorität festlegen

Manchmal ist es hilfreich, wenn die Batchdatei mit einer niedrigen Priorität läuft. Das kann man mittels **start /LOW** erreichen. Weitere Optionen sind **NORMAL**, **HIGH**, **REALTIME**, **ABOVENORMAL** und **BELOWNORMAL**. Das Beispiel zeigt, wie eine Batchdatei quasi sich selber in die niedrige Priorität versetzen kann. In

⁴ [HTTP://GROUPS.GOOGLE.CO.UK/GROUP/ALT.MSDOS.BATCH.NT/MSG/D11E951A822BC420?DMODE=SOURCE](http://groups.google.co.uk/group/alt.msdos.batch.nt/msg/d11e951a822bc420?dmode=source)

dem Fall klappt es allerdings nur, wenn beim ersten Start kein Parameter übergeben wurde.

```
@echo off
if "%1"==" " (
    start /WAIT /LOW /B cmd.exe /V /C %~s0 weiter_machen
    goto:eof
)
echo Jetzt laufe ich mit niedriger Priorität!
echo Überprüfe es im Taskmanager!
pause
```

Funktionsweise: Das Prinzip ist eigentlich ganz einfach. Wenn kein Parameter übergeben wird, wird angenommen, das die Batchdatei zum ersten mal gestartet wurde. Die `if "%1"==" "` Bedingung ist also erfüllt. Mittels **start** wird dann die selbe Batchdatei mit veränderter Priorität gestartet, allerdings mit einem angehängten Parameter **weiter_machen** (Könnte auch irgendwas anderes sein!) Somit ist beim nächsten Aufruf die `if "%1"==" "` Bedingung nicht mehr erfüllt und der normale Teil der Batchdatei wird abgearbeitet.

Wenn man der Batchdatei einen Parameter übergeben möchte (z.B. ein Dateiname o.ä.) muß man alle Parameter verschieben:

```
@echo off
if "%2"==" " (
    start /WAIT /LOW /B cmd.exe /V /C %~s0 %1 weiter_machen
    goto:eof
)
echo Jetzt laufe ich mit niedriger Priorität!
echo Nun kann [%1] 'bearbeitet' werden...
```

pause

Anmerkungen: Normalerweise könnte man statt %~s0 auch "%~0" bei der cmd.exe Zeile verwenden. Allerdings klappt das nicht richtig, wenn Leerzeichen in der Batch Datei vorhanden sind. Mit %~s0 wird der komplette Pfad zur Batchdatei als "Kurznamen" angegeben. In dem Pfad kommt dann keine Leerzeichen vor.

6.0.12 Probleme mit Variablen

Wenn man sich die Hilfeseiten zu **set** (mittels **set /?**) durchliest, stößt man auf das Thema *verzögerte Erweiterung von Variablen*. Das will ich hier mal anhand von Beispielen erklären:

Das Problem

```
set test=1
if "%test%"=="1" (
    set test=2
    echo Wert von 'test' im IF-Block: %test%
)
echo Wert von 'test' nach IF-Block: %test%
```

Man sollte meinen, dass der Wert von %test% in beiden Ausgaben **2** ist. Doch *leider* ist es nicht so. Denn innerhalb des IF-Blocks wird das Neusetzen der Variable test von 1 auf 2 noch nicht aktiv und somit ist das Ergebnis **Wert von 'test' im IF-Block: 1** Erst nach dem IF-Block ist der Wert aktualisiert: **Wert von 'test' nach IF-Block: 2**

Lösung: cmd.exe /V

In einer Batchdatei die mit **cmd /V** gestartet wurde, werden Variablen innerhalb von Befehlsblöcken aktualisiert. Jedoch kann man sie nicht gewohnt mit **%test%** ansprechen, sondern mit **!test!**

```
@echo off
if "%1"==" " (
    start /WAIT /B cmd.exe /V /C "%~0" machen!
    goto:eof
)
set test=1
if "%test%"=="1" (
    set test=2
    echo Wert von 'test'-Prozent in dem IF-Block: %test%
    echo Wert von 'test'-Ausrufezeichen in dem IF-Block: !test!
)
echo Wert von 'test' nach dem IF-Block: %test%
pause
```

6.0.13 Ausgaben/Fehler unterdrücken

Manchmal möchte man per Batch ein Programm starten aber es soll dabei keine Ausgabe gemacht werden. Das ist recht einfach:

```
MeinProgramm.exe >NUL
```

Es könnte aber sein, dass evtl. Fehler dennoch ausgegeben werden. Das liegt daran, dass die Programme in dem Fall auf **stderr** statt **stdout** schreiben. Um auch in dem Fall die Ausgabe zu unter-

drücken, kann man mit einem zusätzlichen `2>&1` die Ausgaben von `stderr` auf `stdout` umleiten. Da `stdout` dann nach **NUL** verschoben wird, sieht man absolut nichts:

```
MeinProgramm.exe >NUL 2>&1
```

6.0.14 Professionelle Message-Fenster erzeugen

Bisher konnte man in Batch keine Fenster erzeugen, bzw. nur in Windows XP mithilfe des Windows Nachrichtendienstes. Doch der ist für Anwendungen viel zu unpraktisch, da man nicht einmal den Fenstertitel bestimmen kann und außerdem ist dieser unter Vista oder Windows 2000 nicht verfügbar.

Ich habe jedoch ein kleines Schlupfloch gefunden, mit dem man dennoch Fenster erzeugen kann:

```
@echo off
Echo msgbox"Text",0,"Fenstername" >Test.vbs
ping localhost -n 3 -w 1000 >NUL
start Test.vbs
pause
```

Erklärung: Man erzeugt hier mithilfe des Umleitungssymbols (Eng.: `pipe`) `>` einen Temporären VBScript, der in der Lage ist, ein Messagefenster zu erzeugen. Dies sorgt in Anwendungen für mehr Übersicht und Professionalität.

6.0.15 Falls das nicht funktioniert

Falls statt eures Textes die Meldung :

"Der Zugriff auf den Windows Scripthost ist auf diesem Computer deaktiviert"... erscheint, dann ist das Öffnen von VBScripts und JScripts aus Sicherheitsgründen verboten.

Um das zu ändern, öffnet ihr den Windows Registrierungseditor (regedit.exe) und löscht NUR folgenden Wert :

HKey_Local_Machine\Software\Microsoft\WindowsScriptHost\Enabled

dann dürfte es funktionieren. **Erstellt vor der Änderung eine Sicherheitskopie eurer Werte !!!**

7 Netzwerkumgebung

7.1 ping

Kann verwendet werden, um die Verbindung zu einem anderen Rechner zu testen. Wenn die Gegenseite antwortet, werden Informationen über die IP-Nummer und zur Verbindungsqualität angezeigt. Zu beachten ist dabei, daß aus einer ausbleibenden Antwort nicht zwangsläufig auf einen mißglückten Verbindungsversuch geschlossen werden darf: manche Administratoren unterbinden diese Art Test aus Sicherheitsgründen.

Beispiel

```
C:\>ping de.wikibooks.org
```

Ausgabe

```
Ping rr.knams.wikimedia.org [145.97.39.155] mit 32 Bytes Daten:

Antwort von 145.97.39.155: Bytes=32 Zeit=67ms TTL=57
Antwort von 145.97.39.155: Bytes=32 Zeit=35ms TTL=57
Antwort von 145.97.39.155: Bytes=32 Zeit=34ms TTL=57
Antwort von 145.97.39.155: Bytes=32 Zeit=39ms TTL=57

Ping-Statistik für 145.97.39.155:
    Pakete: Gesendet = 4, Empfangen = 4, Verloren = 0 (0% Verlust),
    Ca. Zeitangaben in Millisek.:
        Minimum = 34ms, Maximum = 67ms, Mittelwert = 43ms

C:\>
```

Hinweis

Dies lässt sich durch einen Trick mit dem `find` Befehl auch automatisieren:

```
ping de.wikibooks.org | find "TTL" || goto hostnotfound
```

7.2 ipconfig

- Durch den Parameter `/help` erhält man kurze Hilfe.
- Der Aufruf des Befehls `ipconfig` ohne Parameter teilt die eigene IP-Adresse mit.
- Durch den Befehl `ipconfig` lässt sich die IP-Adresse von Netzwerkverbindungen (Ethernetkarten) verwalten.

Beispiel

```
C:\>ipconfig
```

Ausgabe

```
Windows-IP-Konfiguration

Ethernetadapter VMware Network Adapter VMnet8:

    Verbindungsspezifisches DNS-Suffix:
    IP-Adresse. . . . . : 123.456.789.012
    Subnetzmaske. . . . . : 255.255.255.0
    Standardgateway . . . . . :

Ethernetadapter VMware Network Adapter VMnet1:

    Verbindungsspezifisches DNS-Suffix:
    IP-Adresse. . . . . : 123.456.789.1
    Subnetzmaske. . . . . : 255.255.255.0
    Standardgateway . . . . . :
```

```
Ethernetadapter LAN-Verbindung 3:

    Verbindungsspezifisches DNS-Suffix: myhost.local
    IP-Adresse . . . . . : 123.456.789.012
    Subnetzmaske . . . . . : 255.255.255.0
    Standardgateway . . . . . : 192.168.100.1

C:\>
```

7.2.1 ipconfig /all

Eine detailliertere Auskunft erhält man mit dem Parameter /all

Beispiel

```
C:\>ipconfig /all
```

Ausgabe

```
Windows-IP-Konfiguration

    Hostname . . . . . : cestmoi
    Primäres DNS-Suffix . . . . . : mypc.local
    Knotentyp . . . . . : Unbekannt
    IP-Routing aktiviert. . . . . : Nein
    WINS-Proxy aktiviert. . . . . : Nein
    DNS-Suffixsuchliste . . . . . : mypc.local
                                    mypc.local

Ethernetadapter VMware Network Adapter VMnet8:

    Verbindungsspezifisches DNS-Suffix:
    Beschreibung . . . . . : VMware Virtual Ethernet
Adapter for VMnet8
    Physikalische Adresse . . . . . : 01-23-45-67-89-AB
    DHCP aktiviert. . . . . : Nein
    IP-Adresse . . . . . : 123.456.789.1

[ ... ]
```

7.2.2 ipconfig /renew

Sollten die Netzwerkkonfigurationen nicht mehr aufzufinden sein, dann hilft oft der Parameter /renew

Beispiel

```
C:\> ipconfig /renew
```

7.2.3 ipconfig /flushdns

Situation

Eine Homepage wird nicht angezeigt, stattdessen wird eine Fehlermeldung angezeigt, wie z.B. *"Die Website könnte vorübergehend nicht erreichbar sein, versuchen Sie es bitte später nochmals."*

Als Test wird der Server angepingt, und es kommt eine Fehlermeldung

```
Zeitüberschreitung der Anforderung.
```

Der Administrator kümmert sich um das Problem und sagt, der Server sei wieder OK, aber noch immer kommen Fehlermeldungen und Homepages können nicht angezeigt werden. Dann kann es daran liegen, dass der DNS-Cache noch die Fehlermeldung gespeichert hat, und dieser muss geleert werden. Anstelle den Rechner neu zu starten gibt man folgendes ein

```
C:\>ipconfig /flushdns

Windows-IP-Konfiguration
Der DNS-Auflösungscache wurde geleert.
```

```
C:\>
```

7.3 tracert

Mit tracert kann man die Route vom eigenen PC zur Zieladresse verfolgen, und evtl. Rückschlüsse ziehen, ob und warum eine Verbindung hängt.

Beispiel

```
C:\>tracert de.wikibooks.org
```

Ausgabe

```
Routenverfolgung zu rr.knams.wikimedia.org [145.97.39.155] über
maximal 30 Abschnitte:
 1   <1 ms   <1 ms   <1 ms  123.456.789.1
 2   24 ms   18 ms   19 ms  adslgw01-fra4.rm-com.net
[217.173.128.45]
 3  139 ms   35 ms   36 ms  bbcr01-ams.titannetworks.nl
[217.173.143.17]
 4   40 ms   *        43 ms  xsr03.asd002a.surf.net
[195.69.144.34]
 5   41 ms   34 ms   35 ms  AZ-500.XSR01.Amsterdam1A.surf.net
[145.145.80.21]
 6   39 ms   38 ms   37 ms  KNCSW001-router.Customer.surf.net
[145.145.18.158]
 7   41 ms   43 ms   37 ms  gi0-24.csw2-knams.wikimedia.org
[145.97.32.29]
 8   38 ms   38 ms   36 ms  rr.knams.wikimedia.org
[145.97.39.155]
Ablaufverfolgung beendet.
C:\>
```

7.4 netstat

Dieser Befehl zeigt alle offenen TCP/IP/ICMP/UDP (usw...) Verbindungen an.

```
netstat
```

Mit der Erweiterung -p Protokoll lassen sich auch nur Verbindungen zum Entsprechenden Protokoll auslesen(meist TCP):

```
netstat -p TCP
```

7.5 NET

NET ist ein Tool zur Überwachung und Manipulation von Netzwerkaktivitäten sowie Windows-Diensten. Es hat einen recht großen Funktionsumfang, von dem ich hier jedoch nur einige kurz vorstelle.

7.5.1 NET USE - Netzlaufwerke verbinden

NET USE kann genutzt werden um eine Verbindung zu einem Netzwerk- oder Samba-Freigaben herzustellen. Üblicherweise wird hierbei ein Laufwerksbuchstabe verwendet, dies ist jedoch nicht zwingend.

Syntax:

```
NET USE x: \\name.oder.ip.des.rechners\freigabe
```

Wenn für die Verbindung die Eingabe von Zugangsdaten erforderlich ist können diese auch mit angegeben werden:

```
NET USE x: /USER:benutzer \\name.oder.ip.des.rechners\freigabe  
"password"
```

Wenn für einen Rechner Zugangsdaten benötigt werden und diese nicht oder falsch angegeben wurden, so wird der Benutzer nach diesen gefragt. Es kann auch nur ein Benutzer angegeben werden indem das Passwort durch einen Asterisk (*) ersetzt wird.

Die Eingabe von NET USE ohne Parameter zeigt bereits verbundene Laufwerke an. Bei Windows-Freigaben wird auch ein Status angezeigt, bei Samba-Freigaben jedoch nicht.

7.5.2 NET SEND - Nachrichten an andere Rechner senden

Mit NET SEND können kurze Nachrichten an andere Rechner geschickt werden. Der Empfänger sieht diese dann normalerweise in einer Message-Box. Die Benutzung empfiehlt sich nur bedingt, da moderne Firewalls solche Nachrichten normalerweise blocken (siehe Hinweis unten).

Syntax:

```
NET SEND Empfänger Nachricht
```

Empfänger kann dabei sein:

- IP-Adresse
- Domain

- Domain mit Benutzerangabe (/domain:name)
- Rechnername im lokalen Netzwerk
- Benutzername im lokalen Netzwerk
- Asterisk für alle erreichbaren (lokalen) Rechner

Hinweis: Da NET SEND leider sehr oft für Spam missbraucht wird/wurde empfiehlt es sich es abzuschalten. Dies geschieht über den Windows-Service "Nachrichtendienst". Dadurch können weder Nachrichten gesendet noch empfangen werden. (NET STOP Messenger)

7.5.3 NET START

Startet Windows-Dienste

Syntax:

```
NET START Dienstname
```

Die Eingabe von NET START ohne Parameter zeigt die aktuell gestarteten Windows-Dienste an.

7.5.4 NET STOP

Stoppt Windows-Dienste

Gleiche Syntax wie NET START

7.5.5 NET VIEW

Zeigt erreichbare Computer des lokalen Netzwerks, bzw. deren Freigaben an.

Syntax:

```
NET VIEW [\\computername]
```

Einfache Eingabe von NET VIEW zeigt alle Computer an, wenn ein Computername mit angegeben wird werden die Freigaben angezeigt. Dies funktioniert nur bei Windows-Rechnern. UNIX-Rechner mit Samba-Freigaben werden zwar in der Übersicht angezeigt, die Freigaben können aber nicht abgefragt werden.

7.5.6 NET USER

Mit diesem Befehl können Benutzerkonten verwaltet werden. Dazu sind Administratorrechte erforderlich.

Syntax:

```
NET USER
```

Dieser Befehl allein zeigt nur alle Benutzer des Rechners an. (2000) (XP)

```
Net User <name>
```

Durch diesen Befehl wird die Einstellung des Benutzerkontos des Benutzers <name> angezeigt. (2000) (XP)

```
Net User <name> <password> /add
```

Dieser Befehl fügt einen Benutzer hinzu. (2000) (XP)

```
Net User <name> /delete
```

Und dieser Befehl wiederum löscht einen Benutzer. (2000) (XP)

```
Net User <name> <password>
```

Dieser Befehl verändert das Passwort des Benutzers. Wenn anstatt des passworts ein Stern * angegeben wird, dann wird das neue Passwort abgefragt. (2000) (XP)

```
Net User <name> /expires:never
```

Damit wird eingestellt, dass das Passwort des Benutzers <name> niemals abläuft. (2000) (XP)

```
Net User <name> /active:Yes  
Net User <name> /active:No
```

Damit wird das entsprechende Benutzerkonto aktiviert oder deaktiviert. (2000) (XP)

```
Net User <name> /passwordchg:Yes  
Net User <name> /passwordchg:No
```

Damit wird eingestellt, ob der Benutzer sein eigenes Passwort verändern darf. (2000) (XP)

```
Net User <name> /comment:"Herr Maier vom Verkauf"
```

Damit wird der Benutzerkommentar verändert. (2000) (XP)

```
Net User <name> /Fullname:"Karl-Otto Maier-Müller"
```

Damit wird der komplette Benutzername verändert. (2000) (XP)

```
Net User <name> /expires:2009-12-31
```

Damit wird eine Zeit vorgegeben, nach der der Benutzer sein Passwort ändern muss. (2000) (XP)

```
Net User <name> /profilepath="D:\Maier"  
Net User <name> /scriptpath="D:\Maier"  
Net User <name> /homedir="D:\Maier"  
Net User <name> /homedirreq=Yes  
Net User <name> /homedirreq=No
```

Damit wird eingestellt, wo sich die Benutzerdaten befinden. Der Ort der Benutzerdaten wird in der Umgebungsvariable *USER-PROFILE* gespeichert. (2000) (XP)

```
Net User <name> /countrycode=049
```

Damit wird die Ländereinstellung verändert. (2000) (XP)

```
Net User <name> /workstations:*
```

Damit wird es dem Benutzer erlaubt, sich von anderen Rechnern aus an diesem Rechner anzumelden. Anstatt * kann hier auch eine Liste von Computernamen angegeben werden. Dann darf sich der Benutzer nur von diesen Computern anmelden. (2000) (XP)

Hinweis: Diese Arbeitsvorgänge gehen auch mit der Windows Benutzerkontensteuerung. Diese kann erreicht werden durch [Start] [Einstellungen] [Systemsteuerung] [Benutzer]

Hinweis: Der Name des aktuell angemeldeten Benutzers ist in der Umgebungsvariable `UserName`.

Hinweis: Hilfe zum Befehl `Net User` erhält man durch Eingabe von `Net User /help`

Beispiel: Um den Benutzer *Otto* mit dem Passwort *Frieda* einzurichten, kann folgende Batch-Datei ausgeführt werden:

```
net user Otto Frieda /add
net user Otto /expires:never
net user Otto /comment:"Default USER from This Computer"
net user Otto /active:Yes
net user Otto
```

8 Beispiele

8.0.7 Regedit.exe

Eine kleine Batchdatei, um mit regedit.exe direkt einen bestimmten REG-Zweig anzuspringen:

```
set LastKey="HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows
  NT\CurrentVersion\Hotfix"
set Key=HKEY_
CURRENT_
USER\Software\Microsoft\Windows\CurrentVersion\Applets\Regedit
reg add %Key% /f /v "LastKey" /d %LastKey%
start regedit.exe
```

Erläuterung:

- **LastKey** ist der Reg-Schlüssel, der als Letztes angezeigt wurde und dieser wird beim Start von regedit direkt angesprungen, diesen Schlüssel kann man nach Belieben anpassen.
- **Key** ist der Reg-Schlüssel in dem sich der Wert "**LastKey**" befindet.

LastKey wird mittels **reg add** in die Registry eingespielt und danach wird regedit gestartet. Das Ganze funktioniert allerdings nur, wenn regedit.exe noch nicht läuft.

Und hier noch eine kleine Batchdatei, um mit regedit.exe Werte aus der Registry auszulesen:

```
set Key="HKEY_C
URRENT_
USER\Software\Microsoft\Windows\CurrentVersion\Applets\Regedit"
regedit.exe /e c:\temp\inhalt.reg %Key%
```

Achtung! Die Datei inhalt.reg ist im Unicode-Format. Ein Tipp dazu: Der type-Befehl kann Unicode lesen.

8.0.8 TaskList-Abfrage

Eine Routine um das Laufen eines Prozesses mittels **tasklist.exe** festzustellen. In diesem Beispiel wird geprüft, ob **Prozess.exe** läuft.

```
@echo off
set ProgEXE=Prozess.exe
set tempdatei=%temp%\TempDatei.txt
tasklist /FI "IMAGENAME eq %ProgEXE%" /FO CSV>%tempdatei%
for /F "skip=2 tokens=1 delims=" %f in (%tempdatei%) do (
    echo gefiltert: [%f]
    if "%f"=="%ProgEXE%" echo Prozess %ProgEXE% läuft
    goto ende
)
echo Prozess %ProgEXE% läuft nicht!
goto ende
:ende
del %tempdatei%
```

Alternativ, ohne obige temporäre Datei 'tempdatei' zu benötigen, kann man beispielsweise unter Windows XP folgendes verwenden:

```
TaskList /FI "IMAGENAME eq %ProgEXE%" 2>NUL | Find "%ProgEXE%" >NUL
IF ERRORLEVEL == 1 (
    ECHO NOT Running.
    GOTO Ende
```

```
) ELSE (  
    ECHO Running.  
    GOTO Ende  
)
```

8.0.9 Dienste einrichten

```
@echo off  
prompt -$G  
echo.  
echo ### Manuelle Dienste ###  
echo.  
call:M helpsvc          "Hilfe und Support"  
call:M ImapiService     "IMAPI-CD-Brenn-COM-Dienste"  
echo.  
echo ### nicht benötigte Dienste Deaktivieren ###  
echo.  
call:D cisvc            "Indexdienst"  
call:D ERSvc            "Fehlerberichterstattungsdienst"  
echo.  
echo ### Automatische Dienste ###  
echo.  
call:A CryptSvc         "Kryptografiedienste"  
call:A MSIServer        "Windows Installer"  
echo.  
echo.  
echo Fertig!  
echo.  
pause  
goto:eof  
:A  
echo %2 auf Automatisch  
set Parameter=auto  
net start %1  
goto machen  
:M  
echo %2 auf Manuell  
set Parameter=demand  
net stop %1  
goto machen  
:D  
echo %2 auf Deaktiviert  
set Parameter=disabled
```

```
net stop %1
goto machen
rem Systemstart      Gerätetreiber, der vom Startladeprogramm
    geladen wird.
rem system           Gerätetreiber start während der
    Kernelinitialisierung.
rem auto             automatischer start, nach systemstart, vor
    Benutzeranmeldung.
rem demand          Dienst, der manuell gestartet werden muss.
rem disabled        Dienst deaktivieren
:machen
    echo on
    sc config %1 start= %Parameter%
    @echo off
    echo.
    echo.
goto:eof
```

8.0.10 Windows XP SP2: installierte Patches/Updates auflisten

Zeigt unter Windows XP mit Service Pack 2 alle zusätzlich installierten Patches/Updates an.

```
@echo off
set RegHotfixListe=HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows
NT\CurrentVersion\Hotfix
set
  RegHotfixInfo=HKEY_LOCAL_-
MACHINE\SOFTWARE\Microsoft\Updates\Windows
XP\SP3
for /F "delims=\ tokens=7" %%f in ('reg query "%RegHotfixListe%"') do
(
  set HotFixNr=%%f
  CALL :Anzeigen
)
echo.
echo.
pause
goto:eof
:Anzeigen
  reg query "%RegHotfixListe%\%HotFixNr%" /v Installed | FIND
"0x1">NUL
  if "%errorlevel%"=="0" (
```

```
        set installiert=ja
        call:KB_Info
    ) ELSE (
        set installiert=nein
    )
    echo %HotFixNr% - Ist Installiert: %installiert%
    if not "%InstalledBy%"=="    echo %HotFixNr% installiert von
%InstalledBy%
    if not "%InstalledDate%"=="    echo %HotFixNr% am %InstalledDate%
    echo.
goto:eof
:KB_Info
    set InstalledBy=
    for /F "tokens=3" %c in ('reg query "%RegHotfixInfo%\%HotFixNr%"
/v InstalledBy 2^>NUL ^| FIND "InstalledBy"') do (
        set InstalledBy=%c
    )
    set InstalledDate=
    for /F "tokens=3" %c in ('reg query "%RegHotfixInfo%\%HotFixNr%"
/v InstalledDate 2^>NUL ^| FIND "InstalledDate"') do (
        set InstalledDate=%c
    )
goto:eof
```

8.0.11 Printdateien direkt an den Drucker senden

Sendet einzelne Druckdateien an im Netzwerk freigegebene Drucker. Eignet sich besonders bei Standard Drucksprachen wie Postscript und HPGL/2.

```
@echo off
rem "pl.bat" sendet einzelne Druckdateien an bel. Drucker direkt.
rem eignet sich besonders bei Standard Drucksprachen wie Postscript
und HPGL/2.
rem Laesst sich in Windows bequem einbinden, z.B rechte Maustaste -
Senden an Plotter
:voreingestellt
    set server=192.168.0.5
    rem server --- IP von Printserver/oder Name von Freigaberechner
    ---
    set druckerbeschreibung=192.168.0.5 Freigabename EPSON
    rem druckerbeschreibung --- damit der Drucker leichter
```

```
identifiziert werden kann ---
  set drucker=EPSONNT
  rem drucker --- das muss der Freigabename im Netzwerk/auf dem
  Freigaberechner sein ---

  if '%1' == 'drucker01' goto drucker01
  if '%1' == 'drucker02' goto drucker02
  if '%1' == 'canon' goto canon
goto datei
:drucker01
  shift
goto datei
:drucker02
  set server=42.1.1.240
  set druckerbeschreibung=HP DJ 755CM im Sekretariat
  set drucker=hpdj755-tcp
  shift
goto datei
:canon
  set server=192.168.0.1
  set druckerbeschreibung=Canon s/w Laser im EG
  set drucker=CanonLBP
  shift
goto datei
:datei
  if '%~1' == " goto aufruf
goto plot
:plot
  echo.
  echo Dateiname      : %1
  echo Druckserver   : %server%
  echo Freigabename  : %drucker%
  echo Beschreibung  : %druckerbeschreibung%
  echo.
  print /d:\\%server%\\%drucker% '%~1'
goto:eof
:aufruf
echo.
echo  aufruf  : pl [plotter] Datei
echo  plotter : drucker01, drucker02, canon, (voreingestellt ist
drucker01)
echo.
```

Anmerkung zur Nutzung dieses Scriptes:

- Der Printbefehl schickt die Datei OHNE Interpretation direkt an den Drucker. Eine Bilddatei im JPEG-Format würde statt der Grafik also lediglich einen haufen wirrer Zeichen ergeben.
- Der Drucker muß in der Lage sein, selbst ASCII-Zeichen zu verarbeiten und aus zu geben. Dies ist bei älteren Druckern (vor ca. 1996 sicher) immer der Fall. Heute beherrschen meist nur noch PostScript-fähige Drucker diese Fähigkeit. "Billigen" GDI-Druckern muß man etwas auf die Sprünge helfen mit dem verwendeten "Druckprozessor" im Windows-Druckertreiber.

- Zum Umschalten zunächst die Druckerübersicht öffnen - Zu finden unter Start|Drucker und Faxgeräte
- Dann die Eigenschaften des gewünschten Druckers aufrufen (Rechtsklick|Eigenschaften)
- Auf dem Reiter "Erweitert" den Button "Druckprozessor..." klicken
- In dem erscheinenden Dialogfeld gibt es 2 Felder mit Einträgen. Im rechten Feld (Standarddatentypen) findet sich neben verschiedenen anderen auch der Eintrag "Text". Diesen markieren und dann alles mit OK wieder schließen.

Der Druckprozessor von Windows ist nun in der Lage, Ausgaben entsprechend zu Interpretieren und dem GDI-Drucker als Grafik zu liefern. Es können aber nur stur 80 Zeichen (maximal) auf ca. 60 Zeilen verteilt werden. Manchmal sind es bis zu 66 Zeilen.

8.0.12 Copy

Kopieren von Dateien.

Der Befehl **copy test.txt c:\temp** kopiert die Datei *test.txt* aus dem aktuellen Ordner in den Ordner *c:\temp*. Der Befehl **copy**

c:\test.txt c:\temp kopiert die Datei *test.txt* aus dem Ordner *c:* in den Ordner *c:\temp*

Batch-Dateien kennen beim Kopieren nur Ordner mit max. 8+3 Buchstaben deshalb muss man statt z.B.:

```
copy test.txt c:\programme
```

die kurzen Ordner- bzw. Dateinamen

```
copy test.txt c:\progra~1
```

verwenden (ist nicht bei jedem Betriebssystem erforderlich. Im Zweifelsfall ausprobieren.).

```
copy test1.txt + test2.txt test.txt
```

Kombiniert in der gegebenen Reihenfolge zwei oder mehrere durch "+" verbundene Textdateien in einer neuen Gesamtdatei.

In diesem Zusammenhang muss auf die Bedeutung der Parameter "/A" (für ASCII-Format, der Standardwert) und "/B" (für Binär-Format) eingegangen werden, die jedem Dateinamen in obigem Beispiel mit einem Leerzeichen Abstand vorangehen können. DOS-Textdateien enthalten ein Dateienendezeichen (Dezimal 26, Hexadezimal 1A). Beim Kopiervorgang darf in der Gesamtdatei nur als letztes Zeichen ein Dateienendezeichen erscheinen, sonst würde der Text nach der ersten Endmarkierung nicht mehr angezeigt. "copy" kopiert ohne das abschließende Zeichen und fügt als letztes wieder eines zur Gesamtdatei hinzu.

Möchte man Dateien mit binären Inhalten zusammenfügen (z.B. PostScript-Dateien, die auch binäre Abschnitte enthalten kön-

nen), so würde ohne Parameter "/B" ebenfalls nur bis zu einem zufällig vorkommenden Dateiendezeichen kopiert und damit das Ergebnis unbrauchbar.

8.0.13 Message-Ping

Mit Hilfe des folgenden Scripts kann man Windows um eine nützliche Funktionalität erweitern. Mit Hilfe von **mping** erhält man eine Nachricht, sobald der gepingte Computer wieder erreichbar ist.

Um das Script global zu verwenden, muss es z.B. als "mping.cmd" in "**Windows-Installationsordner**\System32" gespeichert werden und kann von nun an z.B. mit "mping google.com" aufgerufen werden.

```
@echo off
:: wenn kein remote Host angegeben ist, dann gehe zu :Syntax
IF [%1]==[] GOTO Syntax
IF [%1]==[/?] GOTO Syntax
IF [%1]==[/h] GOTO Syntax
IF [%1]==[/help] GOTO Syntax
:mainloop
:: pinge den Host 1 mal und leite die Ausgabe nach 'nul' um (keine
  sichtbare Ausgabe)
ping %1 -n 1 >nul
:: speichere den Errorlevel in die Variable 'status'
set status=%errorlevel%
:: sende eine Nachricht sobald der remote Host online ist
if %status%==0 (msg %username% %1 online && exit 0)
:: minimalistische Fortschrittsanzeige
(Set /P i=.) < NUL
:: sleep workaround (wer eine bessere, einfachere, genauere &
  universellere Möglichkeit kennt bitte korrigieren)
:: pinge den lokalen Computer 10 mal
ping localhost -n 10 >nul
:: gehe zurück zum anfang
goto mainloop
:Syntax
echo.
echo Syntax: %~n0 [Hostname oder IP]
```

```
echo.  
echo mping  
echo.  
echo Pingt einen offline Host und gibt eine Meldung  
echo aus, sobald er wieder online ist  
echo.
```

8.0.14 Defrag mit Endlos-Schleife

Das eingebaute Defrag-Tool erlaubt ja immer nur die gleichzeitige Defragmentierung eines Laufwerks. Außerdem sind oftmals mehrere Durchgänge nötig, bis das Laufwerk wirklich vollständig defragmentiert ist. Mit dieser kleinen Batch Datei werden alle Partitionen immer wieder hintereinander defragmentiert. Wenn man wieder arbeiten will, kann man einfach das Eingabefenster schließen. Auch bei laufendem defrag passiert dabei nichts, weil defrag im Hintergrund die aktuelle Arbeit sauber beendet.

Damit aber nicht defekte Laufwerke einfach blind defragmentiert werden, habe ich eine etwas aufwendige Prozedur eingebaut, die die Laufwerke mit chkdsk erst überprüft. Wenn dabei ein Fehler auftritt, wird mit fsutil das dirty Bit gesetzt. Das bewirkt, dass beim nächsten Systemstart die Fehler behoben werden. Außerdem werden die Partitionen in dem loop ausgelassen, welche schon als dirty markiert sind.

Eine kleine Routine war auch nötig, damit Netzwerklaufwerke ausgelassen werden.

```
@echo off  
cd /d C:\  
:loop  
    for %%i in (c,d,e,f,g,h,i,j,k,l,m,n,o,p,q,r,s,t,u,v,w,x,y,z) do (  
        if exist %%i:\NUL call:defrag %%i  
    )  
echo.
```

```
    echo pause...
    ping localhost -n 10 >NUL
goto loop
:defrag
    call:check %1
    if "%ok%"=="nein" (
        echo Laufwerk %1: wird ausgelassen
        echo.
        goto:eof
    )
    echo.
    echo.
    echo chkdsk %1: /v
    chkdsk %1:
    if %errorlevel% NEQ 0 (
        if %errorlevel% NEQ 2 (
            echo %1: chkdsk Fehler!!!
            echo Errorlevel: %errorlevel%
            echo.
            echo setzte dirty-Bit:
            fsutil dirty set %1:
            echo.
            echo Versuche Reparatur
            chkdsk %1: /v /x
            if %errorlevel% EQU 2 (
                echo Fehler bereinigt!
            ) else (
                echo Errorlevel: %errorlevel%
                echo.
                goto:eof
            )
        )
    )
    echo defrag %1: -f
    defrag %1: -f
    if %errorlevel% NEQ 0 (
        pause
        exit
    )
    rem pause...
    ping localhost -n 1 >NUL
goto:eof
:check
    set ok=nein
    net use | find /i "%1:" >NUL
    if not errorlevel 1 (
        echo %1: ist ein Netzlaufwerk
        goto:eof
    )
```

```
)
fsutil dirty query %1:
for /F "tokens=5" %%f in ('fsutil dirty query %1:') do (
    if not "%%f"=="NICHT" (
        rem Laufwerk ist als fehlerhaft markiert
        goto:eof
    )
)
set ok=ja
goto:eof
```

Als Alternative gibt es das Open Source Program JKDEFrag¹, welches auch alle Laufwerke nacheinander defragmentiert.

8.0.15 Dateinformationen anzeigen

Dieses Batch-Skript zeigt Informationen über die Datei, die als Parameter übergeben wird, an. (beispielsweise per Drag&Drop auf die Batch-Datei)

```
@echo off
echo Attribute.....: %~a1
echo Laufwerk.....: %~d1
echo kompletter Pfad.....: %~f1
echo Dateiname.....: %~n1
echo Ordnerpfad.....: %~p1
echo kompletter Pfad (kurze Form): %~s1
echo Aenderungsdatum.....: %~t1
echo Dateierweiterung.....: %~x1
echo Dateigroesse.....: %~z1
pause
```

Hinweis: Falls die Attribute des eigenen Batch-Skripts ausgelesen werden sollen, so muss der Parameter Null verwendet wer-

¹ [HTTP://WWW.KESSELS.COM/JKDEFrag/](http://www.kessels.com/JkDefrag/)

den (Beispielsweise %~p0). Eine Kombination ist auch möglich, z.B: %~dp0 (Laufwerk + Pfad)

8.0.16 Alle verfügbaren PCs im aktuellen Netzwerk suchen

Dieses Batch-Skript pingt alle IP-Adressen eines IP-Bereichs an und zeigt dann die PCs an, die verfügbar sind.

```
@echo off
if exist ips.txt del ips.txt > nul
echo Bitte geben Sie den ersten Teil der IP ein:
set /p ip=
for /L %%N IN (1, 1, 255) DO (
    echo Die IP: %ip%%N wird nun angepingt
    ping %ip%%N -n 1 -w 1 | find "TTL" && echo %ip%%N >> ips.txt
)
cls
type ips.txt
pause >nul
```

Hinweis: Die Datei ips.txt wird nicht gelöscht, damit man sie für weitere Zwecke benutzen kann.

8.0.17 Hosts Datei mit einem Aliasnamen und der aktuellen IP-Adresse aktualisieren

Häufig werden Webserver oder ähnliche Programme unter Angabe eines Aliasnamens installiert, damit sie leicht auf einen anderen Rechner verschoben werden können, falls der ursprüngliche Rechner defekt ist. Oder die Applikation läuft unter VM-Ware auf einen Laptop bei dem sich die IP-Adressen häufig ändern. Dann kann mit dem Skript der Hostnamenalias mit der aktuellen IP-Adresse aktualisiert werden. Idealerweise wird das

Script automatisch beim Start von Windows oder beim Anmelden des Anwenders ausgeführt.

```
@echo off
set HOST_ALIAS=myserver myserver.mydomain.org
set HOST_FILE=%SystemRoot%\System32\drivers\etc\hosts
set TEMP_FILE=%TEMP%\hosts.tmp
REM vorhandenen Eintrag entfernen und den Rest in TEMP-Datei schreiben
type %HOST_FILE% | find /v "myserver" > %TEMP_FILE%
REM Neuen Eintrag wieder hinzufügen
for /F "tokens=15" %i in ('ipconfig ^| find "IP Ad"') do set IP=%i
echo %IP% %HOST_ALIAS%>>%TEMP_FILE%
echo ## Added %IP% %HOST_ALIAS% to the hostsfile
REM Temp-File nach %SystemRoot%\System32\drivers\etc\hosts kopieren
REM Die Datei Hosts darf nicht mit "move" verschoben werde, sondern
  muss kopiert werden, da sonst der Hostname nicht gefunden werden kann.
copy /y %TEMP_FILE% %HOST_FILE%
del /F /Q %TEMP_FILE%
```

8.0.18 Windowsversion herausfinden

Mit dieser Batchdatei findet man heraus, welche Windowsversion verwendet wird. Das ist nützlich, wenn man mit bestimmten Pfaden arbeitet, die je nach Betriebssystem verschieden sind. Hier wird das Ergebnis mithilfe einer VBScript MsgBox ausgegeben.

```
@echo off
VER |find /i "Windows 95" >NUL
IF NOT ERRORLEVEL 1 GOTO 9598ME
VER |find /i "Windows 98" >NUL
IF NOT ERRORLEVEL 1 GOTO 9598ME
VER |find /i "Windows Millennium" >NUL
IF NOT ERRORLEVEL 1 GOTO 9598ME
VER | find "XP" > nul
IF %errorlevel% EQU 0 GOTO XP
VER | find "2000" > nul
IF %errorlevel% EQU 0 GOTO 2000
VER | find "NT" > nul
IF %errorlevel% EQU 0 GOTO NT
VER | find "Microsoft Windows [Version 6" > nul
```

```
IF %errorlevel% EQU 0 GOTO Vista
VER | find "Microsoft Windows [Version 5" > nul
IF %errorlevel% EQU 0 GOTO 2003
goto unknown
goto end
:unknown
echo MsgBox "Betriebssystem unbekannt", VbInformation + VbOKOnly, "Ihr
  Betriebssystem" >%TEMP%\os.vbs
goto end
:9598ME
echo MsgBox "Sie haben 95, 98 oder ME", VbInformation + VbOKOnly, "Ihr
  Betriebssystem" >%TEMP%\os.vbs
goto end
:NT
echo MsgBox "Sie haben Windows NT", VbInformation + VbOKOnly, "Ihr
  Betriebssystem" >%TEMP%\os.vbs
goto end
:2003
echo MsgBox "Sie haben Windows 2003", VbInformation + VbOKOnly, "Ihr
  Betriebssystem" >%TEMP%\os.vbs
goto end
:2000
echo MsgBox "Sie haben Windows 2000", VbInformation + VbOKOnly, "Ihr
  Betriebssystem" >%TEMP%\os.vbs
goto end
:XP
echo MsgBox "Sie haben Windows XP", VbInformation + VbOKOnly, "Ihr
  Betriebssystem" >%TEMP%\os.vbs
goto end
:Vista
echo MsgBox "Sie haben Windows Vista oder Windows 7 RC", VbInformation
  + VbOKOnly, "Ihr Betriebssystem" >%TEMP%\os.vbs
goto end
:end
START %TEMP%\os.vbs
```

8.0.19 TEMP Verzeichnis löschen

Im %TEMP% Verzeichnis sammeln sich mit der Zeit viele Dateien und Unterverzeichnisse an. Mit diesem Script werden alle auf einmal gelöscht:

```
@echo off
call:clean %TEMP%
IF NOT "%TEMP%" == "%TMP%" (
    call:clean %TMP%
)
pause
goto:eof
:clean
del /q "%~1\*.*"
FOR /D %%D IN ("%~1\*") DO (
    rmdir /s /q "%%D"
)
goto:eof
```

(Wahrscheinlich erhält man ein paar Fehler, wenn z.B. Dateien geöffnet sind)

Neue Version eine version für win xp/2000/win7

Echo off

```
del /f /s /q "%HOMEPATH%\Lokale Einstellungen\Temp\*.*"
```

```
del /f /s /q "%HOMEPATH%\Lokale Einstellungen\Temporary Internet Files\*.*"
```

```
del /f /s /q "%HOMEPATH%\Lokale Einstellungen\Verlauf\*.*"
```

```
del /f /s /q "%windir%\Temp\*.*"
```

```
del /f /s /q "%windir%\Prefetch\*.*"
```

```
del /f /s /q "%windir%\Temp\*.*"
```

```
rmdir /s /q "%HOMEPATH%\Lokale Einstellungen\Temp\"
```

```
rmdir /s /q "%windir%\Prefetch\"
```

```
diskperf -n
```

```
ipconfig /flushdns
```

einfach den Text kopieren in eine Textdatei und zur .bat machen
:)

9 Zusatz-Tools

Viele Aufgaben sind mit der Batchprogrammierung bzw. mit Windows-Scripting alleine kaum oder nur sehr umständlich zu lösen. So kann das Umwandeln von Dateien (Textersetzungen) aufgrund der nicht ganz sauberen Behandlung von Sonder- bzw. Operationszeichen zur Qual werden. Erstaunlich für ein Betriebssystem, welches seit nun mehr über *10 Jahren* - mit Windows 95 - diese Restriktionen offiziell beseitigt hat.

Oft können dafür OpenSource-Tools wie SFK ("SWISS FILE KNIFE")¹ und SED² sehr erfolgreich in die Batch-Programmierung eingebunden werden. Für das Verpacken und Komprimieren von Dateien lässt sich das freie 7-ZIP³ gut über die Kommandozeile steuern. Das GPL-Programm Gnuplot eignet sich hervorragend für die Visualisierung von Daten im CSV-Format und lässt sich einfacher und schneller scripten als MS Excel oder OO.org Calc. Zum Herunterladen von Dateien und Webseiten kann Wget verwendet werden. Auch für Batch-Dateien gibt es diverse kostenlose Batch-Compiler, die die Datei in eine ausführbare .exe-Datei umwandeln und den Code vor Manipulation und/oder Copyright-Verletzungen schützen. Für gehobeneres Scripting-Ansprüche und komplexere Vorhaben mit einer längeren Lebensdauer sollten dezidierte Scripting-Sprachen z.B. Awk, PERL⁴, Python, Ruby oder

1 [HTTP://SWISSFILEKNIFE.SOURCEFORGE.NET/](http://swissfileknife.sourceforge.net/)

2 [HTTP://DE.WIKIPEDIA.ORG/WIKI/SED%20%28UNIX%29](http://de.wikipedia.org/wiki/SED%20%28UNIX%29)

3 [HTTP://WWW.7-{ }ZIP.ORG/](http://www.7-zip.org/)

4 [HTTP://DE.WIKIBOOKS.ORG/WIKI/PERL](http://de.wikibooks.org/wiki/Perl)

Tcl in Erwägung gezogen werden, für die glücklicherweise das GNU-Projekt *freie* und quelloffene Alternativen bietet.

Eine kostenlose Alternative zum Batching bietet Microsoft selbst auf seiner Homepage mit dem Tool *PowerShell*.

Weitere Tools sind im Abschnitt "WebLinks" aufgelistet!

10 Referenz

10.1 Liste aller internen und externen Befehle (entspricht der Ausgabe des Befehls `help` unter Windows XP)

ASSOC	Zeigt Dateierweiterungszuordnungen an bzw. ändert sie.
AT	Legt eine Zeit fest, zu der Befehle und Programme auf diesem Computer ausgeführt werden.
ATTRIB	Zeigt Dateiattribute an bzw. ändert sie.
BREAK	Schaltet die erweiterte Überprüfung für STRG+C ein bzw. aus.
CACLS	Zeigt Datei-ACLs (Access Control List) an bzw. ändert sie.
CALL	Ruft eine Batchdatei aus einer anderen Batchdatei heraus auf.
CD	Zeigt den Namen des aktuellen Verzeichnisses an bzw. ändert diesen.
CHCP	Zeigt die aktive Codepagenummer an bzw. legt diese fest.
CHDIR	Zeigt den Namen des aktuellen Verzeichnisses an bzw. ändert diesen.
CHKDSK	Überprüft einen Datenträger und zeigt einen Statusbericht an.

CHKNTFS	Zeigt die Überprüfung des Datenträgers beim Start an bzw. verändert sie.
CLS	Löscht den Bildschirminhalt.
CMD	Startet eine neue Instanz des Windows-Befehlsinterpreters.
COLOR	Legt die Hintergrund- und Vordergrundfarben für die Konsole fest.
COMP	Vergleicht den Inhalt zweier Dateien oder Sätze von Dateien.
COMPACT	Zeigt die Komprimierung von Dateien auf NTFS-Partitionen an bzw. ändert diese.
CONVERT	Konvertiert FAT-Volumes in NTFS. Das aktuelle Laufwerk kann nicht konvertiert werden.
COPY	Kopiert eine oder mehrere Dateien an eine andere Stelle.
DATE	Zeigt das Datum an bzw. legt dieses fest.
DEL	Löscht eine oder mehrere Dateien.
DIR	Listet die Dateien und Unterverzeichnisse eines Verzeichnisses auf.
DISKCOMP	Vergleicht den Inhalt von zwei Disketten.
DISKCOPY	Kopiert den Inhalt von einer Diskette auf eine andere Diskette.
DOSKEY	Bearbeitet Befehlseingaben, ruft Windows-Befehle zurück und erstellt Macros.
ECHO	Zeigt Meldungen an bzw. schaltet die Befehlsanzeige ein oder aus.
ENDLOCAL	Beendet den lokalen Gültigkeitsbereich von Umgebungsänderungen in einer Batchdatei.
ERASE	Löscht eine oder mehrere Dateien.
EXIT	Beendet das Programm CMD.EXE (Befehlsinterpreter).
FC	Vergleicht zwei oder mehr Sätze von Dateien und zeigt die Unterschiede an.

FIND	Sucht eine Zeichenkette in einer oder mehreren Datei(en).
FINDSTR	Sucht Zeichenketten in Dateien.
FOR	Führt einen angegebenen Befehl für jede Datei in einem Dateiensatz aus.
FORMAT	Formatiert einen Datenträger für die Verwendung mit Windows.
FTYPE	Zeigt die Dateitypen an, die bei den Zuordnungen für die entsprechenden Dateierweiterungen verwendet werden bzw. ändert sie.
GETMAC	Zeigt unter Windows XP SP2 die Mac-Adresse an.
GOTO	Setzt den Windows-Befehlsinterpreter auf eine markierte Zeile in einem Batchprogramm.
GRAFTABL	Ermöglicht Windows, Sonderzeichen im Grafikmodus anzuzeigen.
HELP	Zeigt Hilfeinformationen zu Windows-Befehlen an.
IF	Verarbeitet Ausdrücke in einer Batchdatei abhängig von Bedingungen.
LABEL	Erstellt, ändert oder löscht die Bezeichnung eines Volumens.
MD	Erstellt ein Verzeichnis
MKDIR	Erstellt ein Verzeichnis.
MODE	Konfiguriert ein Systemgerät.
MORE	Zeigt Ausgabe auf dem Bildschirm seitenweise an.
MOVE	Verschiebt ein oder mehrere Dateien von einem Verzeichnis in ein anderes.

NET USE	Verbindet einen Computer mit einer freigegebenen Ressource oder trennt die Verbindung und zeigt Informationen über die Verbindungen eines Computers an. Der Befehl steuert außerdem ständige Netzwerkverbindungen.
PATH	Legt den Suchpfad für ausführbare Dateien fest oder zeigt diesen an.
PAUSE	Hält die Ausführung einer Batchdatei an und zeigt eine Meldung an.
POPD	Wechselt zu dem Verzeichnis, das durch PUSHD gespeichert wurde.
PRINT	Druckt eine Textdatei.
PROMPT	Ändert die Eingabeaufforderung.
PUSHD	Speichert das aktuelle Verzeichnis, und wechselt dann zu einem anderen Verzeichnis.
RD	Entfernt ein Verzeichnis.
RECOVER	Stellt lesbare Daten von einem beschädigten Datenträger wieder her.
REM	Leitet Kommentare in einer Batchdatei bzw. CONFIG.SYS ein.
REN	Benennt eine Datei bzw. Dateien um.
RENAME	Benennt eine Datei bzw. Dateien um.
REPLACE	Ersetzt Dateien.
RMDIR	Löscht ein Verzeichnis.
SET	Setzt oder löscht die Umgebungsvariablen bzw. zeigt sie an.
SETLOCAL	Beginnt den lokalen Gültigkeitsbereich von Umgebungsänderungen in einer Batchdatei.
SHIFT	Verändert die Position ersetzbarer Parameter in Batchdateien.
SORT	Sortiert die Eingabe.
START	Startet ein eigenes Fenster, um ein bestimmtes Programm oder einen Befehl auszuführen.

SUBST	Weist einem Pfad einen Laufwerksbuchstaben zu.
TASKLIST	Zeigt alle zurzeit laufenden Aufgaben inklusive der Dienste an.
TASKKILL	Bricht einen laufenden Prozess oder eine Anwendung ab oder beendet ihn bzw. sie.
TIME	Zeigt die Systemzeit an bzw. legt sie fest.
TITLE	Legt den Fenstertitel für das Eingabeaufforderungsfenster fest.
TREE	Zeigt die Ordnerstruktur eines Laufwerks oder Pfads grafisch an.
TYPE	Zeigt den Inhalt einer Textdatei an.
VER	Zeigt die Windows-Version an.
VERIFY	Legt fest, ob überwacht werden soll, ob Dateien korrekt auf den Datenträger geschrieben werden.
VOL	Zeigt die Datenträgervolumenbezeichnung und die Seriennummer an.
XCOPY	Kopiert Dateien und Verzeichnisbäume.

11 Weblinks

11.1 Weblinks

- <http://www.heise.de/ct/03/16/136/default.shtml> c't 16/2003, S. 136: Windows-Befehlszeile
- <http://www.ss64.com/nt/index.html> NT/XP Batch Befehle (en)
- <http://www.admini.de> Umfangreiche Informationen über Batchprogrammierung unter NT/2000/XP
- <http://www.fpschultze.de/> Batch Scripting Site mit vielen Beispielen
- <http://sven-of-nine.de/site/doku.php/faq:batch> Umfangreiche Funktionssammlung für Batchprogrammierer
- <http://www.knowware.de/?cat=6.1&book=batch>
 - <http://download.knowware.de/batch.pdf> Leicht verständliche Einführung
- http://publibz.boulder.ibm.com/cgi-bin/bookmgr_-OS390/BOOKS/IEA2B510/CCONTENTS?SHELF=IEA2BK32&DN=SA22-7598-01&DT=20010626133938 englischsprachiges JCL Benutzerhandbuch
 - <http://www.plogmann.net/w/2/54/index.htm> - Optionen des Befehls XCOPY
- <http://www.antonis.de/dos/#download/> Diverse Beispiel-skripte
- <http://www.easy-coding.de/shellscripting.html?s=5fb492fd4b515b0c880795a> Forum über Shellscripting

- <http://home.mnet-online.de/horst.muc/maind.htm> Eine Sammlung von Zusatztools für Batch-Skriptter
- http://www.axel-hahn.de/axel/page_compi/bat_index.htm Viele weitere Informationen und Tools zum Thema
- http://www.script-example.com/themen/Windows_Batch.php Batch Dateien erstellen für absolute Beginner

11.1.1 Tutorials

- BATCH FOR RUNAWAYS PART I¹ - Tutorial Batch I
- BATCH FOR RUNAWAYS PART II² - Tutorial Batch II
- BATCH FOR RUNAWAYS PART III³ - Tutorial Batch III
- DIE FOR-SCHLEIFE⁴ - Tutorial "FOR"
- VARIABLEN⁵ - Tutorial Umgebungsvariablen

-
- 1 [HTTP://WWW.ADMINISTRATOR.DE/WORKSHOP_BATCH_FOR_RUNAWAYS_-_{}_PART_I_-_{}_BEISPIEL_FINDLONGPATH.BAT_BEDENKLICH_LANGE_PFADE_FINDEN.HTML](http://www.administrator.de/Workshop_Batch_For_Runaways_-_{}_PART_I_-_{}_BEISPIEL_FINDLONGPATH.BAT_BEDENKLICH_LANGE_PFADE_FINDEN.HTML)
 - 2 [HTTP://WWW.ADMINISTRATOR.DE/WORKSHOP_BATCH_FOR_RUNAWAYS_-_{}_PART_II_-_{}_EIN_BISSCHEN_HANDWERKSZEUG.HTML](http://www.administrator.de/Workshop_Batch_For_Runaways_-_{}_PART_II_-_{}_EIN_BISSCHEN_HANDWERKSZEUG.HTML)
 - 3 [HTTP://WWW.ADMINISTRATOR.DE/WORKSHOP_BATCH_FOR_RUNAWAYS_-_{}_PART_III_-_{}_DATUMS_-_{}_UND_ZEITVARIABLEN_IM_BATCH.HTML](http://www.administrator.de/Workshop_Batch_For_Runaways_-_{}_PART_III_-_{}_DATUMS_-_{}_UND_ZEITVARIABLEN_IM_BATCH.HTML)
 - 4 [HTTP://WWW.ADMINISTRATOR.DE/TUTORIAL_ZUR_FOR_-_{}_SCHLEIFE.HTML](http://www.administrator.de/Tutorial_Zur_For_-_{}_Schleife.HTML)
 - 5 [HTTP://WWW.ADMINISTRATOR.DE/WIE_ARBEITE_ICH_MIT_BATCH_UMGEBUNGSVARIABLEN%3F_ERSTELLUNG_UMGANG_ERWEITERUNGEN_VER%C3%A4NDERUNGEN.HTML](http://www.administrator.de/Wie_Arbeite_Ich_Mit_Batch_Umgebungsvariablen%3F_Erstellung_Umgang_Erweiterungen_Ver%C3%A4nderungen.HTML)

11.1.2 Referenzen

* MICROSOFT TECHNET⁶ - Windows Server 2008 - Command Reference (en) * MICROSOFT TECHNET⁷ - Windows Vista - Command-Line Tools Technical Reference (en) * MICROSOFT TECHNET⁸ - Windows Server 2003 - Befehlszeilenreferenz (de) * MICROSOFT TECHNET⁹ - Windows XP - Command-line reference A-Z (en)

-
- 6 [HTTP://TECHNET.MICROSOFT.COM/EN-{}US/LIBRARY/CC754340.ASPX](http://technet.microsoft.com/en-{}us/library/cc754340.aspx)
 - 7 [HTTP://TECHNET.MICROSOFT.COM/EN-{}US/LIBRARY/CC722159.ASPX](http://technet.microsoft.com/en-{}us/library/cc722159.aspx)
 - 8 [HTTP://TECHNET.MICROSOFT.COM/DE-{}DE/LIBRARY/CC785423.ASPX](http://technet.microsoft.com/de-{}de/library/cc785423.aspx)
 - 9 [HTTP://TECHNET.MICROSOFT.COM/EN-{}US/LIBRARY/BB490890.ASPX](http://technet.microsoft.com/en-{}us/library/bb490890.aspx)

12 Autoren

Edits	User
2	ALIZDAIR ¹
1	AQUARIANR ²
2	AUTOR ³
2	AZAËL ⁴
7	BASTIE ⁵
13	BATCHSCRIPTER93 ⁶
2	BLINGNUE ⁷
1	BOERN99 ⁸
10	CODEJUNKIE ⁹

-
- 1 [HTTP://DE.WIKIBOOKS.ORG/W/INDEX.PHP?TITLE=BENUTZER:ALIZDAIR](http://de.wikibooks.org/w/index.php?title=BENUTZER:ALIZDAIR)
 - 2 [HTTP://DE.WIKIBOOKS.ORG/W/INDEX.PHP?TITLE=BENUTZER:AQUARIANR](http://de.wikibooks.org/w/index.php?title=BENUTZER:AQUARIANR)
 - 3 [HTTP://DE.WIKIBOOKS.ORG/W/INDEX.PHP?TITLE=BENUTZER:AUTOR](http://de.wikibooks.org/w/index.php?title=BENUTZER:AUTOR)
 - 4 [HTTP://DE.WIKIBOOKS.ORG/W/INDEX.PHP?TITLE=BENUTZER:Aza%C3%ABL](http://de.wikibooks.org/w/index.php?title=BENUTZER:Aza%C3%ABL)
 - 5 [HTTP://DE.WIKIBOOKS.ORG/W/INDEX.PHP?TITLE=BENUTZER:BASTIE](http://de.wikibooks.org/w/index.php?title=BENUTZER:BASTIE)
 - 6 [HTTP://DE.WIKIBOOKS.ORG/W/INDEX.PHP?TITLE=BENUTZER:BATCHSCRIPTER93](http://de.wikibooks.org/w/index.php?title=BENUTZER:BATCHSCRIPTER93)
 - 7 [HTTP://DE.WIKIBOOKS.ORG/W/INDEX.PHP?TITLE=BENUTZER:BLINGNUE](http://de.wikibooks.org/w/index.php?title=BENUTZER:BLINGNUE)
 - 8 [HTTP://DE.WIKIBOOKS.ORG/W/INDEX.PHP?TITLE=BENUTZER:BOERN99](http://de.wikibooks.org/w/index.php?title=BENUTZER:BOERN99)
 - 9 [HTTP://DE.WIKIBOOKS.ORG/W/INDEX.PHP?TITLE=BENUTZER:CODEJUNKIE](http://de.wikibooks.org/w/index.php?title=BENUTZER:CODEJUNKIE)

- 1 CORNELIUSWASMUND¹⁰
- 1 CREIBENS¹¹
- 7 DANIEL B¹²
- 1 DER LEO¹³
- 20 DIRK HUENNIGER¹⁴
- 1 DONQUICHOT¹⁵
- 1 E(NIX)¹⁶
- 1 ENOMIL¹⁷
- 2 FEHLERKORREKTUR¹⁸
- 2 FLOMINATOR¹⁹
- 21 GENRICH²⁰
- 4 GLOBAL667²¹

-
- 10 [HTTP://DE.WIKIBOOKS.ORG/W/INDEX.PHP?TITLE=BENUTZER:
CORNELIUSWASMUND](http://de.wikibooks.org/w/index.php?title=Benutzer:CorneliusWasmund)
 - 11 [HTTP://DE.WIKIBOOKS.ORG/W/INDEX.PHP?TITLE=BENUTZER:
CREIBENS](http://de.wikibooks.org/w/index.php?title=Benutzer:Creibens)
 - 12 [HTTP://DE.WIKIBOOKS.ORG/W/INDEX.PHP?TITLE=BENUTZER:
DANIEL_B](http://de.wikibooks.org/w/index.php?title=Benutzer:Daniel_B)
 - 13 [HTTP://DE.WIKIBOOKS.ORG/W/INDEX.PHP?TITLE=BENUTZER:
DER_LEO](http://de.wikibooks.org/w/index.php?title=Benutzer:Der_Leo)
 - 14 [HTTP://DE.WIKIBOOKS.ORG/W/INDEX.PHP?TITLE=BENUTZER:
DIRK_HUENNIGER](http://de.wikibooks.org/w/index.php?title=Benutzer:Dirk_Huenniger)
 - 15 [HTTP://DE.WIKIBOOKS.ORG/W/INDEX.PHP?TITLE=BENUTZER:
DONQUICHOT](http://de.wikibooks.org/w/index.php?title=Benutzer:Donquichot)
 - 16 [HTTP://DE.WIKIBOOKS.ORG/W/INDEX.PHP?TITLE=BENUTZER:
E%5E%28NIX%29](http://de.wikibooks.org/w/index.php?title=Benutzer:E%5E%28nix%29)
 - 17 [HTTP://DE.WIKIBOOKS.ORG/W/INDEX.PHP?TITLE=BENUTZER:
ENOMIL](http://de.wikibooks.org/w/index.php?title=Benutzer:Enomil)
 - 18 [HTTP://DE.WIKIBOOKS.ORG/W/INDEX.PHP?TITLE=BENUTZER:
FEHLERKORREKTUR](http://de.wikibooks.org/w/index.php?title=Benutzer:Fehlerkorrektur)
 - 19 [HTTP://DE.WIKIBOOKS.ORG/W/INDEX.PHP?TITLE=BENUTZER:
FLOMINATOR](http://de.wikibooks.org/w/index.php?title=Benutzer:Flominator)
 - 20 [HTTP://DE.WIKIBOOKS.ORG/W/INDEX.PHP?TITLE=BENUTZER:
GENRICH](http://de.wikibooks.org/w/index.php?title=Benutzer:Genrich)
 - 21 [HTTP://DE.WIKIBOOKS.ORG/W/INDEX.PHP?TITLE=BENUTZER:
GLOBAL667](http://de.wikibooks.org/w/index.php?title=Benutzer:Global667)

- 1 HADDOCK²²
- 1 HARDY42²³
- 5 HEULER06²⁴
- 4 HJHERBERT²⁵
- 2 JAN²⁶
- 1 JODDEL²⁷
- 13 JUETHO²⁸
- 1 KLARTEXT²⁹
- 16 KLAUS EIFERT³⁰
- 1 KLEIDERSACK³¹
- 1 KOMIKAA³²
- 6 KRONOS86M³³

-
- 22 [HTTP://DE.WIKIBOOKS.ORG/W/INDEX.PHP?TITLE=BENUTZER:HADDOCK](http://de.wikibooks.org/w/index.php?title=Benutzer:Haddock)
 - 23 [HTTP://DE.WIKIBOOKS.ORG/W/INDEX.PHP?TITLE=BENUTZER:HARDY42](http://de.wikibooks.org/w/index.php?title=Benutzer:Hardy42)
 - 24 [HTTP://DE.WIKIBOOKS.ORG/W/INDEX.PHP?TITLE=BENUTZER:HEULER06](http://de.wikibooks.org/w/index.php?title=Benutzer:Heuler06)
 - 25 [HTTP://DE.WIKIBOOKS.ORG/W/INDEX.PHP?TITLE=BENUTZER:HJHERBERT](http://de.wikibooks.org/w/index.php?title=Benutzer:Hjherbert)
 - 26 [HTTP://DE.WIKIBOOKS.ORG/W/INDEX.PHP?TITLE=BENUTZER:JAN](http://de.wikibooks.org/w/index.php?title=Benutzer:Jan)
 - 27 [HTTP://DE.WIKIBOOKS.ORG/W/INDEX.PHP?TITLE=BENUTZER:JODDEL](http://de.wikibooks.org/w/index.php?title=Benutzer:Joddel)
 - 28 [HTTP://DE.WIKIBOOKS.ORG/W/INDEX.PHP?TITLE=BENUTZER:JUETHO](http://de.wikibooks.org/w/index.php?title=Benutzer:Juetho)
 - 29 [HTTP://DE.WIKIBOOKS.ORG/W/INDEX.PHP?TITLE=BENUTZER:KLARTEXT](http://de.wikibooks.org/w/index.php?title=Benutzer:KlarText)
 - 30 [HTTP://DE.WIKIBOOKS.ORG/W/INDEX.PHP?TITLE=BENUTZER:KLAUS_EIFERT](http://de.wikibooks.org/w/index.php?title=Benutzer:Klaus_Eifert)
 - 31 [HTTP://DE.WIKIBOOKS.ORG/W/INDEX.PHP?TITLE=BENUTZER:KLEIDERSACK](http://de.wikibooks.org/w/index.php?title=Benutzer:Kleidersack)
 - 32 [HTTP://DE.WIKIBOOKS.ORG/W/INDEX.PHP?TITLE=BENUTZER:KOMIKAA](http://de.wikibooks.org/w/index.php?title=Benutzer:Komikaa)
 - 33 [HTTP://DE.WIKIBOOKS.ORG/W/INDEX.PHP?TITLE=BENUTZER:KRONOS86M](http://de.wikibooks.org/w/index.php?title=Benutzer:Kronos86M)

- 7 MF-WARBURG³⁴
- 5 MICHAELFREY³⁵
- 2 MICHAELFREYTOOL³⁶
- 7 MICHFRM³⁷
- 11 MJCHAE³⁸
- 1 NOMIKE³⁹
- 3 OCRHO⁴⁰
- 2 PC-WORLD⁴¹
- 1 PEPERKORN⁴²
- 1 PRINCE KASSAD⁴³
- 1 PRODUNIS⁴⁴
- 6 RESEKA⁴⁵

34 [HTTP://DE.WIKIBOOKS.ORG/W/INDEX.PHP?TITLE=BENUTZER:MF-WARBURG](http://de.wikibooks.org/w/index.php?title=Benutzer:MF-WARBURG)

35 [HTTP://DE.WIKIBOOKS.ORG/W/INDEX.PHP?TITLE=BENUTZER:MICHAELFREY](http://de.wikibooks.org/w/index.php?title=Benutzer:MICHAELFREY)

36 [HTTP://DE.WIKIBOOKS.ORG/W/INDEX.PHP?TITLE=BENUTZER:MICHAELFREYTOOL](http://de.wikibooks.org/w/index.php?title=Benutzer:MICHAELFREYTOOL)

37 [HTTP://DE.WIKIBOOKS.ORG/W/INDEX.PHP?TITLE=BENUTZER:MICHFRM](http://de.wikibooks.org/w/index.php?title=Benutzer:MICHFRM)

38 [HTTP://DE.WIKIBOOKS.ORG/W/INDEX.PHP?TITLE=BENUTZER:MJCHAE](http://de.wikibooks.org/w/index.php?title=Benutzer:MJCHAE)

39 [HTTP://DE.WIKIBOOKS.ORG/W/INDEX.PHP?TITLE=BENUTZER:NOMIKE](http://de.wikibooks.org/w/index.php?title=Benutzer:NOMIKE)

40 [HTTP://DE.WIKIBOOKS.ORG/W/INDEX.PHP?TITLE=BENUTZER:OCRHO](http://de.wikibooks.org/w/index.php?title=Benutzer:OCRHO)

41 [HTTP://DE.WIKIBOOKS.ORG/W/INDEX.PHP?TITLE=BENUTZER:PC-WORLD](http://de.wikibooks.org/w/index.php?title=Benutzer:PC-WORLD)

42 [HTTP://DE.WIKIBOOKS.ORG/W/INDEX.PHP?TITLE=BENUTZER:PEPERKORN](http://de.wikibooks.org/w/index.php?title=Benutzer:PEPERKORN)

43 [HTTP://DE.WIKIBOOKS.ORG/W/INDEX.PHP?TITLE=BENUTZER:PRINCE_KASSAD](http://de.wikibooks.org/w/index.php?title=Benutzer:PRINCE_KASSAD)

44 [HTTP://DE.WIKIBOOKS.ORG/W/INDEX.PHP?TITLE=BENUTZER:PRODUNIS](http://de.wikibooks.org/w/index.php?title=Benutzer:PRODUNIS)

45 [HTTP://DE.WIKIBOOKS.ORG/W/INDEX.PHP?TITLE=BENUTZER:RESEKA](http://de.wikibooks.org/w/index.php?title=Benutzer:RESEKA)

- 3 ROBERT⁴⁶
- 10 ROBIDOG⁴⁷
- 25 SCHNEIJO⁴⁸
- 1 SE4598⁴⁹
- 1 STEFAN MAJEWSKY⁵⁰
- 1 STEFAN WICHMANN⁵¹
- 1 STEFANTIRAN⁵²
- 2 SYS.BAK⁵³
- 4 THEPACKER⁵⁴
- 10 TIMON.FREITAG⁵⁵
- 1 TSCHÄFER⁵⁶
- 2 UWE.KAUSCH⁵⁷

-
- 46 [HTTP://DE.WIKIBOOKS.ORG/W/INDEX.PHP?TITLE=BENUTZER:ROBERT](http://de.wikibooks.org/w/index.php?title=BENUTZER:ROBERT)
 - 47 [HTTP://DE.WIKIBOOKS.ORG/W/INDEX.PHP?TITLE=BENUTZER:ROBIDOG](http://de.wikibooks.org/w/index.php?title=BENUTZER:ROBIDOG)
 - 48 [HTTP://DE.WIKIBOOKS.ORG/W/INDEX.PHP?TITLE=BENUTZER:SCHNEIJO](http://de.wikibooks.org/w/index.php?title=BENUTZER:SCHNEIJO)
 - 49 [HTTP://DE.WIKIBOOKS.ORG/W/INDEX.PHP?TITLE=BENUTZER:SE4598](http://de.wikibooks.org/w/index.php?title=BENUTZER:SE4598)
 - 50 [HTTP://DE.WIKIBOOKS.ORG/W/INDEX.PHP?TITLE=BENUTZER:STEFAN_MAJEWSKY](http://de.wikibooks.org/w/index.php?title=BENUTZER:STEFAN_MAJEWSKY)
 - 51 [HTTP://DE.WIKIBOOKS.ORG/W/INDEX.PHP?TITLE=BENUTZER:STEFAN_WICHMANN](http://de.wikibooks.org/w/index.php?title=BENUTZER:STEFAN_WICHMANN)
 - 52 [HTTP://DE.WIKIBOOKS.ORG/W/INDEX.PHP?TITLE=BENUTZER:STEFANTIRAN](http://de.wikibooks.org/w/index.php?title=BENUTZER:STEFANTIRAN)
 - 53 [HTTP://DE.WIKIBOOKS.ORG/W/INDEX.PHP?TITLE=BENUTZER:SYS.BAK](http://de.wikibooks.org/w/index.php?title=BENUTZER:SYS.BAK)
 - 54 [HTTP://DE.WIKIBOOKS.ORG/W/INDEX.PHP?TITLE=BENUTZER:THEPACKER](http://de.wikibooks.org/w/index.php?title=BENUTZER:THEPACKER)
 - 55 [HTTP://DE.WIKIBOOKS.ORG/W/INDEX.PHP?TITLE=BENUTZER:TIMON.FREITAG](http://de.wikibooks.org/w/index.php?title=BENUTZER:TIMON.FREITAG)
 - 56 [HTTP://DE.WIKIBOOKS.ORG/W/INDEX.PHP?TITLE=BENUTZER:TSCH%C3%A4FER](http://de.wikibooks.org/w/index.php?title=BENUTZER:TSCH%C3%A4FER)
 - 57 [HTTP://DE.WIKIBOOKS.ORG/W/INDEX.PHP?TITLE=BENUTZER:UWE.KAUSCH](http://de.wikibooks.org/w/index.php?title=BENUTZER:UWE.KAUSCH)

- 2 WEISSNIX⁵⁸
- 1 WOLFGANGBEYER⁵⁹
- 3 WORKER⁶⁰

58 [HTTP://DE.WIKIBOOKS.ORG/W/INDEX.PHP?TITLE=BENUTZER:WEI%C3%9FNIX](http://de.wikibooks.org/w/index.php?title=Benutzer:Wei%C3%9Fnix)

59 [HTTP://DE.WIKIBOOKS.ORG/W/INDEX.PHP?TITLE=BENUTZER:WOLFGANGBEYER](http://de.wikibooks.org/w/index.php?title=Benutzer:Wolfgangbeyers)

60 [HTTP://DE.WIKIBOOKS.ORG/W/INDEX.PHP?TITLE=BENUTZER:WORKER](http://de.wikibooks.org/w/index.php?title=Benutzer:Worker)

13 Bildnachweis

In der nachfolgenden Tabelle sind alle Bilder mit ihren Autoren und Lizenzen aufgelistet.

Für die Namen der Lizenzen wurden folgende Abkürzungen verwendet:

- GFDL: Gnu Free Documentation License. Der Text dieser Lizenz ist in einem Kapitel dieses Buches vollständig angegeben.
- cc-by-sa-3.0: Creative Commons Attribution ShareAlike 3.0 License. Der Text dieser Lizenz kann auf der Webseite <http://creativecommons.org/licenses/by-sa/3.0/> nachgelesen werden.
- cc-by-sa-2.5: Creative Commons Attribution ShareAlike 2.5 License. Der Text dieser Lizenz kann auf der Webseite <http://creativecommons.org/licenses/by-sa/2.5/> nachgelesen werden.
- cc-by-sa-2.0: Creative Commons Attribution ShareAlike 2.0 License. Der Text der englischen Version dieser Lizenz kann auf der Webseite <http://creativecommons.org/licenses/by-sa/2.0/> nachgelesen werden. Mit dieser Abkürzung sind jedoch auch die Versionen dieser Lizenz für andere Sprachen bezeichnet. Den an diesen Details interessierten Leser verweisen wir auf die Onlineversion dieses Buches.

- cc-by-sa-1.0: Creative Commons Attribution ShareAlike 1.0 License. Der Text dieser Lizenz kann auf der Webseite <http://creativecommons.org/licenses/by-sa/1.0/> nachgelesen werden.
- cc-by-2.0: Creative Commons Attribution 2.0 License. Der Text der englischen Version dieser Lizenz kann auf der Webseite <http://creativecommons.org/licenses/by/2.0/> nachgelesen werden. Mit dieser Abkürzung sind jedoch auch die Versionen dieser Lizenz für andere Sprachen bezeichnet. Den an diesen Details interessierten Leser verweisen wir auf die Onlineversion dieses Buches.
- cc-by-2.0: Creative Commons Attribution 2.0 License. Der Text dieser Lizenz kann auf der Webseite <http://creativecommons.org/licenses/by/2.0/deed.en> nachgelesen werden.
- cc-by-2.5: Creative Commons Attribution 2.5 License. Der Text dieser Lizenz kann auf der Webseite <http://creativecommons.org/licenses/by/2.5/deed.en> nachgelesen werden. Mit dieser Abkürzung sind jedoch auch die Versionen dieser Lizenz für andere Sprachen bezeichnet. Den an diesen Details interessierten Leser verweisen wir auf die Onlineversion dieses Buches.
- cc-by-3.0: Creative Commons Attribution 3.0 License. Der Text dieser Lizenz kann auf der Webseite <http://creativecommons.org/licenses/by/3.0/deed.en> nachgelesen werden. Mit dieser Abkürzung sind jedoch auch die Versionen dieser Lizenz für andere Sprachen bezeichnet. Den an diesen Details interessierten Leser verweisen wir auf die Onlineversion dieses Buches.
- GPL: GNU General Public License Version 2. Der Text dieser Lizenz kann auf der Webseite

<http://www.gnu.org/licenses/gpl-2.0.txt> nachgelesen werden.

- PD: This image is in the public domain. Dieses Bild ist gemeinfrei.
- ATTR: The copyright holder of this file allows anyone to use it for any purpose, provided that the copyright holder is properly attributed. Redistribution, derivative work, commercial use, and all other use is permitted.
- EURO: This is the common (reverse) face of a euro coin. The copyright on the design of the common face of the euro coins belongs to the European Commission. Authorised is reproduction in a format without relief (drawings, paintings, films) provided they are not detrimental to the image of the euro.
- LFK: Lizenz Freie Kunst. Der Text dieser Lizenz kann auf der Webseite <http://artlibre.org/licence/lal/de> nachgelesen werden.
- CFR: Copyright free use. Der Urheberrechtsinhaber erlaubt es jedem, dieses Bild für jeglichen Zweck, inklusive uneingeschränkter Weiterveröffentlichung, kommerziellem Gebrauch und Modifizierung, zu nutzen.
- EPL: Eclipse Public License. Der Text dieser Lizenz kann auf der Webseite <http://www.eclipse.org/org/documents/epl-v10.php> nachgelesen werden.

Bild	Autor	Lizenz
------	-------	--------