



# Batch-Infos

Copyright: www.admini.de

| <u><a href="#">Gesammelte Script-Beispiele (Batch, VBS etc.)</a></u>      |   |  |
|---|---|--|
| <u><a href="#">Allgemeines zu Batchdateien</a></u>                        | <u><a href="#">Ausführen / Starten von Batchdateien</a></u>                     | <u><a href="#">Aufruf von externen Befehlen</a></u>              |
| <u><a href="#">Konfiguration + Tipps zur DOS-Box</a></u>                  | <u><a href="#">Verwenden von Sprungmarken (Labels)</a></u>                      | <u><a href="#">Arbeiten mit Umgebungsvariablen</a></u>           |
| <u><a href="#">Parameterübergabe an Batchdateien und Unterrouinen</a></u> | <u><a href="#">Befehlssymbole (Sonderzeichen mit speziellen Funktionen)</a></u> | <u><a href="#">Automatische Batch-Ausführung</a></u>             |
| <u><a href="#">Unbeaufsichtigte Installationen (Unattended setup)</a></u> | <u><a href="#">Fehlersuche / Debugging</a></u>                                  | <u><a href="#">Batch-Fehlermeldungen + Mögliche Ursachen</a></u> |
| <u><a href="#">Regeln für gute Batchprogramme</a></u>                     | <u><a href="#">Weitere Informationen / Links zur Batchprogrammierung</a></u>    |  |

## Allgemeines zu Batchdateien

- Batchdatei = Textdatei mit **ASCII- (DOS-) Zeichensatz**, die eine Liste von Windows-Befehlen oder anderen ausführbaren Dateien enthält, die bei Aufruf der Batchdatei Zeile für Zeile abgearbeitet werden
- Dateierweiterung wahlweise **".bat"** (läuft unter allen DOS- und Windows-Versionen) oder **".cmd"** (läuft nur NT 4.0 / 2000 / XP)
- jeder Befehl darf maximal eine Zeile umfassen (kein freier Text wie unter C / Pascal u.a.)
- Groß-/Kleinschreibung der Befehle, Variablennamen etc. sind i.A. egal (engl. "case-insensitive")  
Ausnahmen:
  - IF-Befehl (ohne Parameter /i):  
Beim Vergleich von zwei Zeichenfolgen mit "=", EQU und NEQ wird standardmäßig zwischen Groß- und Kleinschreibung unterschieden - z.B. IF XP==xp echo Zeichenfolgen sind nicht gleich!
  - FIND- und FINDSTR-Befehl (ohne Parameter /i):  
auch hier wird die in Anführungszeichen gesetzte Zeichenfolge standardmäßig case-sensitive gesucht: FIND "windows xp" findet nicht die Zeichenfolge "Windows XP"
  - FOR-Befehl:  
die Variablen %a bis %z sind nicht identisch mit den Variablen %A bis %Z (= insgesamt 52 Variablen verfügbar)

- einige aus dem UNIX-Bereich übernommene Befehle unterscheiden zwischen groß- und kleingeschriebenen Parametern, z.B. FTP -a ... oder FTP -A ...
- Leerzeilen und Zeilen, die mit dem Befehl "REM" beginnen, werden bei der Programmabarbeitung ignoriert
- Bearbeiten von Batchdateien mit einem beliebigen Text-Editor möglich, der Windows- bzw. DOS-Zeichensatz unterstützt (nicht UNICODE !) - z.B. Notepad, DOS-Edit oder auch Proton Code Editor (siehe unten - "Weitere Informationen / Links zur Batchprogrammierung")
- **Umlaute** problematisch - daher möglichst weglassen oder DOS-Edit verwenden (oder WinWord + *Speichern unter* - Dateityp "MS-DOS-Text")
- Aufruf einer Batchdatei erfolgt unter NT/2000/XP standardmäßig immer über cmd.exe (Kommandointerpreter) = eigene DOS-Box je Batchdatei
- Jede Zeile der Batchdatei wird einzeln in die DOS-Box "geschrieben" - wobei Variablen ggf. durch ihren Wert ersetzt werden - und dann ausgeführt; somit wird standardmäßig jede ausgeführte Befehlszeile, die nicht mit dem Zeichen @ (AT bzw. Klammeraffe) beginnt, angezeigt. Dies lässt sich jedoch mit dem Befehl "@echo off" abschalten; üblicherweise steht dieser Befehl daher am Anfang jeder Batchdatei.

## Ausführen / Starten von Batchdateien

| Aufruf  | Parameterübergabe möglich ?                                       | Fenster automatisch geschlossen ?<br>* | Aktuelles (Arbeits-) Verzeichnis                      |
|---|---|--|---|
| durch Doppelklick im Explorer   | Nein  | Ja                                     | = Speicherort der Batchdatei                          |
| durch Ziehen einer beliebigen Datei im Explorer auf eine Batchdatei (Drag&Drop)                         | nur Dateiname/Pfad der gezogenen Datei                            | Ja                                     | = Speicherort der Batchdatei                          |
| über <i>Start - Ausführen</i>   | Ja  | Ja                                     | unklar (z.T. C:\ oder Benutzerprofil)                 |
| über DOS-Eingabeaufforderung + Eingabe des Dateinamens der Batchdatei mit absolutem oder relativem Pfad | Ja  | Nein                                   | kann über CD-Befehl festgelegt werden                 |
| Batchdatei aus dem Explorer in eine geöffnete DOS-Eingabeaufforderung ziehen (Drag&Drop)                | Ja  | Nein                                   | kann über CD-Befehl festgelegt werden                 |
| Verknüpfung (.lnk-Datei), z.B. im Startmenü   | Ja (siehe Befehlszeile / Verknüpfungseigenschaften / Feld "Ziel") | Ja                                     | siehe Verknüpfungseigenschaften / Feld "Ausführen in" |
| u.a.  |   |  |   |

\* = evtl. Fehlermeldungen oder sonstige Bildschirmausgaben u.U. nicht sichtbar

## Aufruf von Befehlen und Programmen aus einer Batchdatei heraus:

- Unterscheidung zwischen **internen Befehlen**, die in cmd.exe enthalten sind und **externen Befehle**, die als eigenständige .Exe-Dateien etc. abgelegt sind (z.B. xcopy, sort, etc.)  
Liste interner Befehle: assoc, call, chdir/cd, cls, color, copy, date, dir, dpath, echo, endlocal, erase/del, exit, for, ftype, goto, if, mkdir/md, move, path, pause, popd, prompt, pushd, rem, rename/ren, rmdir/rd, set, setlocal, shift, start, time, title, type, ver  
 (vgl. <http://www.microsoft.com/technet/winnt/shellscr.asp?a=frame#e>)
- Aufruf von **Windows-Befehlen** ohne Pfad möglich, da die dazugehörigen .EXE-Dateien unter c:\windows\system32 gespeichert sind und dieser Pfad in der PATH-Variablen enthalten ist
- Aufruf **anderer Befehle, Tools sowie beliebiger Anwendungen** normalerweise nur mit Pfadangabe (oder stattdessen PATH-Variable erweitern - siehe Kapitel Umgebungsvariablen bzw. PATH-Befehl)
- Aufruf von beliebigen **Konsolenbefehlen** möglich, die ausschließlich innerhalb der bereits geöffneten DOS-Box arbeiten (siehe auch Windows-Befehle CMD und START)
- Aufruf von **Befehlen / Programmen mit grafischer Oberfläche (GUI)** aus einer Batchdatei bzw. aus der DOS-Box heraus möglich (siehe auch START-Befehl)
- Aufruf von **16- oder 32-Bit-Programmen** möglich.  
 Beim Aufruf von 16-Bit-Anwendungen dürfen Name + Pfad der ausführbaren Datei nicht länger als 128 Zeichen sein; ebenso darf die gesamte Befehlszeile einschließlich Name der ausführbaren Datei (aber ohne Pfad !) nicht länger als 130 Zeichen sein !  
 (vgl. Microsoft Knowledgebase Artikel Q198429 - PRB: Executing a 16-Bit Process with a Large Command Line on NT)
- **Nichtausführbare Dateien** können einfach durch Eingabe des Namens als Befehl aufgerufen werden, wenn der entsprechende Dateityp einer Anwendung zugeordnet ist (z.B. würde WORD.DOC die Anwendung starten, die der .DOC-Erweiterung zugeordnet ist, also winword.exe) - siehe auch FTYPE und ASSOC sowie im Explorer unter *Ansicht - Optionen - Dateitypen*.

## Konfiguration + Tipps zur DOS-Box (= Eingabeaufforderung / Konsolenfenster)

- Die Konfiguration der DOS-Box bei Aufruf über *Start - Ausführen* - "cmd.exe" richtet sich nach den Einstellungen in den Dateien C:\WINDOWS\\_DEFAULT.PIF, %SystemRoot%\SYSTEM32\AUTOEXEC.NT und %SystemRoot%\SYSTEM32\CONFIG.NT (sofern die Datei \_DEFAULT.PIF nicht geändert wurde).
- Bei Aufruf über *Start - Programme - Eingabeaufforderung* richtet sich die Konfiguration nach der entsprechenden Verknüpfung im Startmenü (.LNK-Datei).
- Die **Eigenschaften der DOS-Box** lassen sich über *Systemsteuerung - Konsole* über das Systemmenü der DOS-Box definieren (Alt + Leertaste).  
 Empfehlenswerte Einstellungen:
  - Optionen "Quick-Edit-Modus" und "Einfüge-Modus" aktivieren (Details siehe dortige Online-Hilfe-Schaltfläche)
  - Farben: schwarze Schrift auf weißem Hintergrund
  - *Layout - Fensterpuffergröße - Höhe*: 300 - hierdurch erscheint ein vertikaler Scrollbalken ("Bildlaufleiste"), so dass auch bereits vom Bildschirm verschwundene Zeilen wieder

eingebildet werden kann (siehe auch MORE-Befehl)

Diese Einstellungen müssen 2x vorgenommen werden:

1. beim Aufruf der DOS-Box über *Start - Ausführen - cmd.exe* Systemmenü aufrufen, Einstellungen ändern und beim Schließen die (+ Option "Eigenschaften gelten für alle Fenster mit gleichem Namen" wählen) (oder über *Systemsteuerung - Konsole*) und
2. bei Aufruf über *Start - Programme - Zubehör - DOS-Eingabeaufforderung* Systemmenü aufrufen, Einstellungen ändern und beim Schließen die (+ Option "Verknüpfung, die dieses Fenster aufruft, ändern" wählen)

Die entsprechenden Einstellungen werden in der Registry unter HKCU\Console + ggf. Unterschlüsseln (je nach Titel der DOS-Box) gespeichert.

- **Kopieren von Text** aus der DOS-Box in die Zwischenablage (hierfür muss der QuickEdit-Modus aktiviert sein - siehe oben):
  - über Systemmenü (Alt + Leertaste - Bearbeiten - Markieren - anschließend gewünschten Text mit Umschalt + Pfeiltasten markieren und mit ENTER in die Zwischenablage kopieren
  - oder über Maus (ggf. vorher Quick-Edit-Modus aktivieren (siehe oben); dann Text mit Maus markieren und mit ENTER in die Zwischenablage kopieren)
- **Einfügen von Text** aus der Zwischenablage in die DOS-Box:
  - Mausklick mit der rechten Maustaste
  - oder über Tastaturkürzel Alt+Leertaste + anschließend "b" (Bearbeiten) + "e" (Einfügen)
- **Einfügen von Dateinamen / Pfadangaben** aus dem Explorer in die DOS-Box:
  - mit Hilfe von **Drag&Drop vom Explorer in die DOS-Box** lassen sich bequem Datei- oder Verzeichnisnamen "schreiben": Gewünschte Datei im Explorer suchen, mit der Maus ein Stück zur Seite ziehen, mit Alt + Tab zur DOS-Box umschalten und Maustaste in diesem Fenster loslassen
  - oder mit Hilfe des Freeware-Tools PfadName.exe: hiermit lässt sich das Explorer-Kontextmenü um Einträge zum Kopieren von Dateinamen / Pfad der ausgewählten Datei in die Zwischenablage erweitern; der kopierte Pfad lässt sich dann bequem in der DOS-Box bzw. in der Batchdatei (Notepad etc.) wieder einfügen (siehe <http://www.groebe-software.de/freeware/programme/ppfadname.htm>)
- Verwenden des Pseudo-Laufwerks B: (= zweite Diskette im gleichen Diskettenlaufwerk / unter MS-DOS erschien z.B. bei DIR B: die Meldung "... legen Sie eine Diskette in Laufwerk B:") - unter NT/2K/XP nur noch beim Befehl DISKCOPY möglich
- **Wiederholen einer Befehlszeile** mit den Tasten Pfeil nach oben / unten (DOSKEY.EXE nicht erforderlich)
- **Verwerfen einer Befehlszeile** (Eingabe abbrechen) mit der ESC-Taste
- Beim Wechseln in ein Verzeichnis mit Hilfe des **CD-Befehls** ist das Jokerzeichen "\*" sehr nützlich, z.B. durch Eingabe von "cd C:\Pro\*" + Enter statt "cd C:\Programme"
- Über den Registryschlüssel HKCU\Software\Microsoft\Command Processor\**CompletionChar** kann eine Taste - üblicherweise die Tabulatortaste (DWord-Wert 00000009) - festgelegt werden, mit der sich ein angefangener Dateiname **automatisch vervollständigen** lässt, z.B. durch Eingabe von "type c:\windows\s" + Tabulatortaste; durch mehrmaliges Drücken kann zwischen mehreren Dateien / Verzeichnissen gewechselt werden, die mit dieser Zeichenfolge beginnen oder einem Muster entsprechen (z.B. type c:\windows\\*.ini + Tabulatortaste).
- Mit Hilfe des Befehls DOSKEY /HISTORY > c:\temp\batch.cmd lässt sich bequem eine Batchdatei mit den zuletzt in der DOS-Box eingegebenen Befehlen erzeugen.

- siehe auch Infos zu dem Befehl CMD

## Verwenden von Sprungmarken (Labels)

- Sprungmarken dienen nur zur Adressierung bei Goto- und Call-Befehlen
- Sprungmarke = Zeichenfolge + vorangestellter Doppelpunkt
- Empfehlung: max. 20 Zeichen, keine Umlaute, Sonderzeichen, etc.
- Sprungmarken stehen immer am Zeilenanfang
- Zeilen mit Sprungmarken werden bei der Abarbeitung der Batchdatei übersprungen
- System-Sprungmarke :EOF ist vordefiniert; bewirkt das Verlassen der Batchdatei bzw. der aktuellen Unterroutine (wenn vorher eine Unterroutine mit CALL aufgerufen wurde)
- Einzelheiten: siehe GOTO und CALL

## Arbeiten mit Umgebungsvariablen

- maximale Länge der Namen von Umgebungsvariablen: theoretisch unbegrenzt (bis 600 Zeichen getestet)  
Empfehlung: max. 20 Zeichen, keine Umlaute, Sonderzeichen, etc., da sonst z.T. Probleme auftreten können
- Groß-/Kleinschreibung im Variablennamen wird nicht berücksichtigt (case-insensitive)
- Definition von Variablen über SET-Befehl oder SETX (Resource-Kit)
- Variablen sind alphanumerisch, d.h. sie enthalten grundsätzlich nur eine Folge von ASCII-/Ansi-Zeichen (bei Berechnung eines arithmetischen Ausdrucks (SET mit /a) wird dieser Ausdruck nur temporär während der Berechnung in einen numerischen Wert umgewandelt)
- Verwendung in Batchdatei durch Einschließen mit Prozentzeichen, z.B. echo %Eingabe%  
**Ausnahme:** i.V.m. SET /a Variablennamen ohne Prozentzeichen ! (siehe SET-Befehl)
- einfache **Stringmanipulationen** bei Umgebungsvariablen möglich (aber nicht bei sonstigen Variablen wie %1, %a, etc.) - d.h. Zerlegen von Zeichenfolgen, Rückgabe einzelner Zeichen und Suchen/Ersetzen von Zeichenfolgen (Details - siehe SET-Befehl)
- Verwendung von Variablen (mit wenigen Ausnahmen) im gesamten Windows-System möglich (z.B. beim Einrichten eines neuen Symbols / Programmaufrufen etc.)
- Ausführen von Kommandos unter Einsatz von Umgebungsvariablen möglich, z.B. lässt sich der Befehl "notepad c:\autoexec.bat" als Umgebungsvariable "aeb" speichern und kann dann durch Eingabe von "%aeb%" bzw. durch eine entsprechende Zeile in der Batchdatei aufgerufen werden.

### Unterschiede zwischen verschiedenen Typen von Umgebungsvariablen

- Windows unterscheidet zwischen **Benutzervariablen** (User Environment), **Systemvariablen** (Machine Environment) und **temporären Umgebungsvariablen** (Volatile Environment), die alle sowohl gelesen als auch verändert werden können.  
Darüber hinaus gibt es noch **interne Variablen** (z.B. %ComputerName%, %SystemRoot%, etc.), die zwar gelesen aber nicht (direkt) verändert werden können.

- Benutzervariablen werden in der **Registry** unter HKCU\Environment (also in der ntuser.dat), Systemvariablen unter HKLM\SYSTEM\CurrentControlSet\Control\Session Manager\Environment und temporäre Variablen unter HKCU\Volatile Environment gespeichert. Manuelle Änderungen in der Registry werden allerdings erst bei der nächsten Benutzeranmeldung bzw. durch einen speziellen Systemaufruf wirksam (siehe Microsoft Knowledgebase - Article ID: Q104011).
- Innerhalb der DOS-Box lässt sich mit "SET <Variable>=<Wert>" nur eine temporäre Variable erzeugen, die beim Beenden der DOS-Box verloren geht. Wird dieser Befehl jedoch in der Datei C:\AUTOEXEC.BAT ausgeführt, legt Windows diese Variable als interne Variable an! Das gleiche gilt entsprechend für den PATH-Befehl.
- **Kollision mehrerer Variablentypen:** mit "SET <Variable>" lässt sich der Wert einer einzelnen Umgebungsvariablen ermitteln, egal ob Benutzer-, System-, temporäre oder interne Variable; falls mehrere gleichnamige Variablen definiert wurden, haben temporäre Variablen Vorrang vor Benutzervariablen und diese wiederum Vorrang vor Systemvariablen (d.h. wenn die Variable TEST im User Environment und im Machine Environment existiert, wird der Wert der Benutzervariablen TEST angezeigt).  
Ausnahme: PATH-Variable - hier werden ggf. die Werte mehrerer vorhandener Variablen automatisch durch Semikolon getrennt hintereinander gehängt. (siehe PATH-Befehl), dies geschieht auch automatisch bei Verwendung des NetwareBefehls "MAP" in Verbindung mit Suchlaufwerken
- Anzeigen und Definieren von System- und Benutzervariablen über *Systemsteuerung - System - Umgebung* möglich (Windows 2000: *Systemsteuerung - System - Erweitert - Umgebungsvariablen*)
- Mit SETX.EXE aus dem Resource Kit lassen sich Benutzer- und System-Umgebungsvariablen erzeugen, die dauerhaft gespeichert werden.  
Da das in der Registry abgelegte Environment (Benutzer-/Systemvariablen) nur einmalig beim Starten einer DOS-Box ausgelesen wird, werden nach Aufruf der DOS-Box mit SETX erzeugte bzw. geänderte Variablen nicht aktualisiert; die Änderungen werden also erst für darauffolgende Aufrufe der DOS-Box wirksam. Um eine Variable sofort **und** dauerhaft zur Verfügung zu stellen, müssen daher SET und SETX kombiniert verwendet werden.

```

set var=abcdefg
    REM temporäre Variable "var" anlegen
setx var %var%
    REM temporäre Variable ins User Environment kopieren, und
    REM damit dauerhaft zur Verfügung stellen; stattdessen
    könnte
    REM auch der Befehl "setx var abcdefg" verwendet werden

```

- Im Netware-Loginscript mit dem Befehl "DOS SET" definierte (DOS-) Variablen werden (dauerhaft) als interne Variablen gespeichert.

- Liste der verfügbaren Umgebungsvariablen siehe auch Kapitel "Übersicht über die Eingabeaufforderung" / Abschnitt "Verwenden von Umgebungsvariablen mit Cmd.exe"
- siehe auch Kapitel "Erweiterte Funktionen für Batch-Parameter und Batch-Variablen"
- siehe auch: SET-Befehl
- Weitere Infos im Internet unter <http://www.ss64.demon.co.uk/nt/set.html>

## Parameterübergabe an Batchdateien und Unterroutinen

Hinweis: Die folgenden Anmerkungen beziehen sich v.a. auf die Parameterübergabe an Batchdateien, gelten aber sinngemäß auch für die Parameterübergabe an beliebige Windows-Befehle sowie an Unterroutinen (siehe CALL-Befehl).

- Parameter = Zeichenfolgen in der Befehlszeile, die auf den Namen der ausführbaren Datei folgen
- als Parameter können z.B. Dateinamen/-pfade, Schalter (z.B. /?) und sonstige Informationen übergeben werden
- einzelne Parameter werden jeweils durch Leerzeichen, Semikolon oder Komma getrennt
- Jeder Parameter kann innerhalb der Batchdatei mit Hilfe der Variablen %1 bis %9 abgefragt werden; %\* gibt alle Parameter zurück (siehe auch CALL / GOTO sowie SHIFT).
- Mit Hilfe der Variablen %0 lässt sich der Dateiname der Batchdatei so wie er in der Befehlszeile angegeben wurde, abfragen (z.B. mit/ohne absolutem/relativem Pfad, mit/ohne Erweiterung etc.), d.h. %0 enthält exakt die Befehlszeile aber ohne Parameter.  
Falls die Batchdatei aus dem Explorer heraus mit Doppelklick gestartet wurde, beinhaltet %0 den Dateinamen mit komplettem (absolutem) Pfad (z.B. c:\windows\test.cmd); mit %0\.. lässt sich der Dateiname aus dem Pfad herauslösen (z.B. c:\windows).  
Um den Dateinamen mit vollständigem Pfad unabhängig vom Aufruf der Batchdatei zu ermitteln kann „%~f0“ verwendet werden (siehe CALL- / GOTO-Befehl).
- Für die Parameter %0 bis %9 sowie für die beim FOR-Befehl verfügbaren Variablen %a bis %z bietet Windows **spezielle Pfadoperatoren**, um einen Dateinamen in seine Bestandteile zu zerlegen, also den gesamten absoluten Pfad oder nur Teile davon (Laufwerksbuchstabe, Verzeichnis, Dateiname, Dateierweiterung) zu ermitteln - Einzelheiten: siehe CALL-Befehl
- Soll eine Zeichenfolge, die Leerzeichen enthält, als **ein** Parameter übergeben werden, muss die gesamte Zeichenfolge in Anführungszeichen gesetzt werden, z.B. xcopy "c:\windows\all users\\*.lnk" c:\backup
- **Weitere Infos im Internet unter <http://www.ss64.demon.co.uk/ntsyntax/parameters.html> sowie unter <http://www.ss64.demon.co.uk/ntsyntax/esc.html>**

## Befehlssymbole (Sonderzeichen mit speziellen Funktionen)

➤ siehe auch Kapitel "Verwenden von Befehlsumleitungsoperatoren" und "Verwenden von Filtern"

**&** Befehlsverkettung; ermöglicht mehrere Befehle in einer Zeile (entspricht „:“ unter BASIC)

**&&** bedingte Befehlsverkettung - nachfolgender Befehl wird nur ausgeführt, wenn der vorhergehende Befehl erfolgreich war, also die Variable %ERRORLEVEL% nach dessen Ausführung den Wert 0 hat.

**Achtung:** Dies kann aber auch dann zutreffen, wenn der betreffende Befehl zwar fehlgeschlagen ist, aber keinen ERRORLEVEL setzt ! - z.B.:

```
@echo off
```

```
xcopy abc def
```

```
REM XCOPY-Befehl setzt Errorlevel auf 4, da Datei  
"abc"
```

```
REM nicht vorhanden
```

`demo.exe && echo demo.exe erfolgreich ausgeführt`

Falls der Befehl "demo.exe" keinen ERRORLEVEL setzt (d.h. diesen unverändert lässt), wird der darauffolgende Echo-Befehl nur ausgeführt, wenn der vorhergehende XCOPY-Befehl erfolgreich war, was aber in diesem Beispiel nicht der Fall ist

**||** bedingte Befehlsverkettung - nachfolgender Befehl wird nur ausgeführt, wenn der unmittelbar zuvor ausgeführte Befehl fehlgeschlagen ist, also die Variable %ERRORLEVEL% nach dessen Ausführung einen Wert ungleich 0 hat.

**Achtung:** Dies kann aber auch dann zutreffen, wenn der betreffende Befehl zwar erfolgreich war, aber keinen ERRORLEVEL setzt (und ERRORLEVEL vorher bereits ungleich 0 war) !

**Sonderfall:** **||** i.V.m. CALL - z.B. CALL :Sub **||** echo Fehler !

In diesem Fall wird der ECHO-Befehl nicht nur ausgeführt, wenn der CALL-Befehl einen Fehler verursacht, sondern auch, wenn der letzte Befehl innerhalb der Unterroutine einen ERRORLEVEL setzt !

(siehe auch Hinweise zu IF- und GOTO-Befehl)

**( und )** Klammern dienen zur Gruppierung, d.h. zur logischen Zusammenfassung von Befehlen oder von mehreren Befehlszeilen (siehe Hinweise zu IF-, FOR- und GOTO-Befehl)

**;** oder **,** oder Leerzeichen

Befehlsparameter voneinander trennen (siehe Kapitel "Parameterübergabe an Batchdateien")

**"** (in doppelte Anführungszeichen eingeschlossene Zeichenfolge)

Übergabe einer Zeichenfolge als 1 Befehlsparameter, wenn die Zeichenfolge Leerzeichen, Semikolon oder Komma enthält (siehe Kapitel "Parameterübergabe an Batchdateien")

**^** Escape-Zeichen - bewirkt, dass nachfolgendes Sonderzeichen bzw. Befehlssymbol als normaler Text interpretiert wird (z.B. echo ^> um das Zeichen „>“ auszugeben statt die Ausgabe in eine Datei umzuleiten; siehe auch FOR-Befehl)

**|** (Pipe-Symbol)

dient dazu, die Ausgabe eines Befehls als Eingabe eines 2. Befehls zu verwenden (Befehlsverkettung)

**/** wird meist Befehlszeilenschaltern vorangestellt (z.B. xcopy /?) - siehe auch Kapitel "Parameterübergabe an Batchdateien")

**< und >** Ermöglichen die Umleitung von Bildschirm- und Ausgaben auf eine Datei oder ein Gerät (z.B. CON, LPT1, COM1, PRN oder NUL (= Ausgabe "wegwerfen") - siehe auch MODE-Befehl):

**0>** bzw. **<**

= Umleitung der Standard-Eingabe (normalerweise Gerät CON = Tastatur); jede Benutzereingabe wird statt von der Tastatur aus einer Datei (oder von der seriellen Schnittstelle) gelesen.

**>** oder **1>**

= Umleitung der Standard-Ausgabe, also der normalen Bildschirmausgabe eines Befehls in eine Datei oder auf ein Gerät (Standard: Gerät CON = Bildschirm). Bei Umleitung auf eine Datei wird hierbei eine neue Datei erzeugt bzw. eine ggf. bereits existierende Datei überschrieben.

**>>** oder **1>>**

= Umleitung der Standard-Ausgabe. Bei Umleitung auf eine Datei wird hierbei eine neue Datei erzeugt bzw. die Ausgabe an eine ggf. existierende Datei angehängt

2>

= Umleitung der Standard-Fehlerausgabe, also nur der Ausgabe von Fehlermeldungen eines Befehls (normalerweise Gerät CON = Bildschirm). Bei Umleitung auf eine Datei wird hierbei eine neue Datei erzeugt bzw. eine ggf. bereits existierende Datei überschrieben.

2>>

= Umleitung der Standard-Fehlerausgabe. Bei Umleitung auf eine Datei wird hierbei eine neue Datei erzeugt bzw. die Ausgabe an eine ggf. existierende Datei angehängt

1>test.log 2>>&1

= Umleitung der Standard-Ausgabe in die Datei test.log + Umleitung der Standard-Fehlerausgabe auf die (umgeleitete) Standard-Ausgabe (also ebenfalls test.log).

- Siehe auch IF-Befehl - "Beispiele für die Verwendung des IF-Befehls in Verbindung mit Befehlsverkettungen"

## Automatische Batch-Ausführung unter Windows

Um eine Batchdatei oder ein beliebiges Programm bei der Benutzeranmeldung automatisch aufrufen zu lassen gibt es verschiedene Möglichkeiten:

- über eine beliebige Befehlszeile in der Datei C:\AUTOEXEC.BAT
- mit Hilfe des Windows- oder Netware-Anmeldescripts des Benutzers
- mit Hilfe des Registry-Schlüssels „HKEY\_CURRENT\_USER\Software\Microsoft\Windows\CurrentVersion\Load“ (?)
- ab Windows 2000 beim Aufruf der DOS-Box mit Hilfe der Registry-Schlüssel "Software\Microsoft\Command Processor\AutoRun" und "Software\Microsoft\Command Processor\AutoRun" (HKLM + HKCU)
- oder über verschiedene weitere Registry-Schlüssel, die in einem Artikel aus der Microsoft Knowledgebase erläutert werden (Article ID Q179365) - hier ein Auszug daraus:

Under Windows NT 4.0, the keys are loaded in the following order:

```
HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\RunOnceEx
HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\RunOnce
HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run
StartUp Folder
```

(AutoStart-Ordner - Ausführung lässt sich durch Drücken der Umschalt-Taste nach der Anmeldung unterdrücken)

HKEY\_CURRENT\_USER\Software\Microsoft\Windows\CurrentVersion\RunOnce

With the exception of the HKEY\_LOCAL\_MACHINE\...\RunOnce key, all keys and their entries are loaded asynchronously. Therefore, all entries in the RunServices and RunServicesOnce keys can potentially run at the same time.

Entries in the HKEY\_LOCAL\_MACHINE\...\RunOnce key are loaded synchronously in an undefined order.

Because the HKEY\_LOCAL\_MACHINE\...\RunOnce key is loaded synchronously, all of its entries must finish loading before the HKEY\_LOCAL\_MACHINE\...\Run, HKEY\_CURRENT\_USER\...\Run, HKEY\_CURRENT\_USER\...\RunOnce, and Startup Folder entries can be loaded.

The data value for a value is a command line. The value name can be prefixed with an exclamation point (!) to defer deletion of the value until after the command has been completed.

Der von MS-DOS / Win3x her bekannte Eintrag in der WIN.INI (Abschnitt [Windows] / Parameter Run=...) funktioniert dagegen unter NT/2K/XP offensichtlich nicht mehr.

## Unbeaufsichtigte Installationen (unattended setup / Silent Installation)

### 1. MSI-Installationen (Microsoft Installer)

siehe <http://www.microsoft.com/TechNet/prodtechnol/winxppro/proddocs/msiexec.asp>,  
<http://unattended.sourceforge.net/installers.html>

### 2. InstallShield-Setups

Installationen lassen sich meist über eine angepasste setup.iss + entsprechende Aufrufparameter automatisieren - siehe <http://www.swynk.com/trent/Articles/ISSCommands.asp>,  
<http://www.installshield.com>, sowie <http://unattended.sourceforge.net/installers.html>

*Setup.exe is the main InstallShield executable; it performs setup initialization and launches the appropriate InstallShield engine file (\_instxxx.ex\_) to execute the setup script (Setup.ins) on the target system. You must ship this file on Disk1 of your distribution disks. If desired, you can rename the program file and distribute it freely with its new name.*

*Command line parameters*

*Following is the list of command line parameters that can be used with Setup.exe. These switches are optional, but note that either a slash (/) or a dash (-) must precede the command line parameters. Separate multiple command line parameters with a space. But do not put a space inside a command line parameter (for example, /r /fInstall.ins is valid, but not /r/f Install.ins).*

*When using long path and file name expressions with switches, enclose the expressions in double quotation marks. The enclosing double quotes tell the operating system that spaces within the quotation marks are not to be treated as command line delimiters.*

*Setup.exe command line parameters are not case sensitive; upper- or lowercase letters can be used.*

*You can pass command line parameters directly to Setup.exe, or you can place the command line parameters in Setup.ini.*

*Parameter Description*

*-f<path\CompiledScript>*

*The alternate compiled script can be specified using this option. Unless the compiled script (.ins file) also resides in the same directory as that of Setup.exe, the full path to the compiled script must be specified. \_setup.dll must also reside in the same directory as your .ins file. For example, setup -ftest.ins will launch setup using Test.ins instead of Setup.ins.*

*-f1<path\ResponseFile>*

*The alternate location and name of the response file (.iss file) can be specified using this option. If this option is used when running InstallShield Silent, the response file is read from the folder/file specified by <path\ResponseFile>. If an alternate compiled script is specified using the -f switch, the -f1 switch entry must follow the -f switch entry.*

*-f2<path\LogFile>*

*The alternate location and name of the log file created by InstallShield Silent can be specified by using this option. By default, Setup.log log file is created and stored in the same directory as that of Setup.exe. If an alternate compiled script is specified using the -f switch, the -f2 switch entry must follow the -f switch entry.*

*-d*

*Run setup in debug mode. The -d switch also includes a <pathonly> option for specifying the path of the Setup.rul file. For more information, refer to the Visual Debugger help file.*

*-m<file name>*

*Causes InstallShield to generate a Management Information Format (.mif) file automatically at the end of the installation. Do not include a path - the .mif file is always placed in the Windows folder. <file name> is optional. If you do not specify a file name, the resulting file will be called Status.mif.*

*-m1<serial number>*

*Tells InstallShield to place the indicated serial number in the created .mif file.*

*-m2<locale string>*

*Tells InstallShield to place the indicated locale in the .mif file. English (ENU) is the default; refer to Microsoft documentation for a complete listing of locale strings.*

*-r*

*Causes Setup.exe automatically to generate a silent installation file (.iss file), which is a record of the installation input, in the Windows folder.*

*-s*

*Run InstallShield Silent to execute a silent installation.*

*-SMS*

*The SMS switch prevents a network connection and the Setup.exe from closing before the installation is complete. This switch works with installations originating from a Windows NT server over a network. Please note that SMS must be uppercase; this is a case-sensitive switch.*

## Additional Information

### User-defined command line parameters

Along with the command line parameters listed above, -c, -e, -q, -t, -x, -z and -z1, are command line parameters reserved by InstallShield. User redefinition of these command line parameters, both upper- and lowercase, can cause errors. The user defined command line parameters must be specified before specifying the predefined Setup options, if they are applicable

Quelle: <http://www.swynk.com/trent/Articles/ISSCommands.asp>

### 3. Wise Installer Setups

siehe [www.wise.com](http://www.wise.com)

### 4. Sonstige Setups

- Informationen zu automatisierter Installation in readme.txt o.a. Dokumentation mitgeliefert ?
- Informationen zu automatisierter Installation über setup /?
- Informationen zu automatisierter Installation über setup /ü (==> liefert u.U. Fehlermeldung "Falscher Schalter '/ü' " + Informationen zu möglichen Parametern)
- Aufruf von setup.exe mit /q bzw. -q (für "quiet") oder /s bzw. -s (für "silent") testen
- siehe auch <http://unattended.sourceforge.net/installers.html>

### 5. Setups, die keine automatisierten Installationen unterstützen

- evtl. Steuerung der grafischen Oberfläche des Setups (GUI) + Beantwortung der Fragen / Auswahl der Installationsoptionen mit Hilfe von ScriptIt (MS-Freeware - siehe <http://www.microsoft.com/technet/treeview/default.asp?url=/TechNet/prodtechnol/winntas/downloads/scriptit.asp>) oder AutoIt (Freeware - siehe <http://www.hiddensoft.com/AutoIt/>) möglich
- siehe auch <http://unattended.sourceforge.net/installers.html>, <http://www.autoit.com>

## Fehlersuche / Debugging

### **MÖGLICHE PROBLEME:**

1. Fehlermeldung am Bildschirm (siehe Kapitel "Batch-Fehlermeldungen + Mögliche Ursachen")
2. Unerwarteter Abbruch der Batchdatei (ohne Fehlermeldung)
3. Fehlerhafte Ausgaben, Berechnungen, etc.
4. Fehlerhafte Verzweigungen
5. Sonstige Fehler

### **LÖSUNGSANSÄTZE:**

#### 1. Fehlerhafte Zeile auffinden:

- Befehlsanzeige einschalten: echo on
- Zusatzinformationen ausgeben lassen: Inhalt von Variablen, Inhalt von Verzeichnissen, beliebige Texte (um die fehlerhafte Zeile einzugrenzen), etc.
- Batch-Abarbeitung mit PAUSE unterbrechen lassen
- Batchdatei mit Ausgabeumleitung aufrufen, um alle Bildschirmausgaben (inkl. Fehlermeldungen) protokollieren zu lassen, z.B. testbat.cmd 1>testbat.log 2>>&1

## 2. Fehlerursache herausfinden / beheben

- Syntax des betreffenden Befehls überprüfen
- ggf. Inhalt der verwendeten Variablen überprüfen
- ggf. Vorhandensein und Inhalt der verwendeten Dateien und Verzeichnisse überprüfen
- bei Verwendung relativer Pfade: aktuelles Verzeichnis überprüfen (z.B. mit "cd" ohne Parameter)

## 3. Falls Fehlerursache nicht nachvollziehbar ist:

- Fehlerhafte Batchdatei kopieren + nach und nach Teile herauslöschen und immer wieder testen, so lange, bis der Fehler abgestellt ist. Auf diese Weise lässt sich auch herausfinden, ob der Fehler gar nicht allein mit der angenommenen Befehlszeile zusammenhängt.  
(Beispiel: Ein Befehl funktioniert nur deshalb nicht einwandfrei, weil zuvor mit CD in ein anderes (falsches) Verzeichnis gewechselt wurde, was bei der Fehlersuche leicht übersehen werden kann.)
- Bei Datei-/Verzeichnisoperationen: Betreffende Aktion zunächst auf einem anderen Laufwerk / mit anderen Dateien bzw. Verzeichnissen testen.
- Fehlerhafte Befehlszeilen umformulieren, umbauen oder mit anderer Syntax oder anderen Befehlen probieren.  
(Beispiel: falls XCOPY Fehlermeldung erzeugt, COPY oder unter Netware NCOPY testen, etc.)
- Kollegen/innen fragen (kein Scherz - manchmal ist man nach längerer Arbeit am PC einfach "betriebsblind" !)
- **Achtung: Nicht alle Fehler sind auf Programmierfehler zurück zu führen - teilweise ist auch Windows fehlerhaft ("buggy") !**

Beispiel: Beim Aufruf einer 2. Batchdatei aus einer Unteroutine heraus erscheint die Fehlermeldung, dass die Sprungmarke der Unteroutine nicht gefunden wird.

In solchen Fällen ist meist ein wenig Phantasie gefragt: Im o.g. Beispiel lässt sich das Problem ganz leicht lösen, indem der START-Befehl vor den Namen der aufzurufenden Batchdatei geschrieben wird.

## 4. Mögliche Fehlerursachen / beliebte Fehler:

- Voraussetzungen der Batchdatei nicht erfüllt (z.B. bestimmte Verzeichnisse oder Dateien, die existieren müssen; Variablen, die gesetzt sein müssen; Registry-Einträge; freier Speicherplatz, etc.)
- mangelnde Zugriffsrechte (Workstation + Server)
- PC- / Benutzerabhängige Fehler (Batchdatei auf anderem PC / unter anderer Benutzeranmeldung testen)
- aktuelles Verzeichnis falsch; z.B. weil CD-Befehl ohne Parameter /d verwendet wurde !

- TEMP-Variable nicht gesetzt (kann z.B. Fehlermeldung bei Filterbefehlen verursachen)
- Dateien schreibgeschützt oder mit Attribut "System" oder "Versteckt" versehen
- Syntax-Fehler - meist durch Schreibfehler oder fehlerhafte Angabe der Parameter eines Befehls verursacht
- IF-Befehl: Vergleich mit einfachem statt doppeltem Gleichheitszeichen (z.B. if %1=/y goto ... )
- IF-Befehl: Vergleich ohne Berücksichtigung von Groß- und Kleinschreibung (siehe IF-Befehl / Parameter /i)
- IF-Befehl: Vergleich mit einer Variablen oder einem Ausdruck, der einen leeren Wert liefert (= Syntaxfehler / siehe auch Hinweise zu IF-Befehl)
- Endlosschleifen (z.B. durch "goto :Start" + fehlende Abbruchbedingung)
- fehlende Sprungmarken / doppelt definierte Sprungmarken / fehlender Doppelpunkt bei CALL-Befehl (call sub), Schreibfehler, etc.
- fehlender GOTO-Befehl am Ende des Hauptprogramms, wenn darunter noch Unterroutinen stehen (==> nach Abarbeitung des Hauptprogramms wird zusätzlich noch die 1. Unterroutine gestartet - allerdings ohne Aufruf mit CALL !)
- Probleme mit Umlauten / Unterschiede zwischen Windows-/DOS-Zeichensatz (Notepad / Edit) - z.B. Pfad "c:\windows\profiles\administrator\programme\startmenü" (siehe Kapitel "Allgemeines zu Batchdateien)
- Pfade nicht in Anführungszeichen gesetzt (problematisch, wenn Leerzeichen etc. enthalten sind - z.B. "c:\windows\profiles\all users")
- einige Befehle haben eine etwas abweichende Funktion, je nachdem, ob sie direkt in der DOS-Box oder über eine Batchdatei aufgerufen werden (siehe Hinweise zu entsprechenden Befehlen)

## Batch-Fehlermeldungen + Mögliche Ursachen

| Batch-Fehler  | Mögliche Ursachen  |
|---|--|
| Fehlermeldung "Der Befehl ist entweder falsch geschrieben oder konnte nicht gefunden werden. Bitte überprüfen Sie die Schreibweise und die Umgebungsvariable 'PATH'." | <p>→ Fehlermeldung kann sich auf den Befehl am Zeilenanfang beziehen oder auf einen Befehl der einem der folgenden Sonderzeichen folgt:</p> <p>     &amp; &amp;&amp;</p> <ul style="list-style-type: none"> <li>• Befehl falsch geschrieben</li> <li>• absolute Pfadangabe der ausführbaren Datei fehlerhaft</li> <li>• relative Pfadangabe der ausführbaren Datei fehlerhaft oder aktuelles Verzeichnis falsch</li> </ul> |
| Fehlermeldung "Das Sprungziel - xxx wurde nicht gefunden."  | <ul style="list-style-type: none"> <li>• Sprungmarke in einem GOTO- oder CALL-Befehl falsch geschrieben oder nicht definiert</li> </ul>  |

|   |  |
|---|--|
|   | <ul style="list-style-type: none"> <li>• Sprungmarke in einem CALL-Befehl ohne vorangestellten Doppelpunkt (z.B. call abc)</li> <li>• Sprungmarke ":EOF" ohne vorangestellten Doppelpunkt verwendet (z.B. goto eof)</li> </ul>   |
| Fehlermeldung "Das System kann die angegebene Datei nicht finden."  | <ul style="list-style-type: none"> <li>• Befehlszeile enthält versehentlich ein Umleitungszeichen o.ä. (z.B. echo Syntax: test &lt;Verzeichnisname&gt;)</li> </ul>   |
| Fehlermeldung "Der Prozeß kann nicht auf die Datei zugreifen, da sie von einem anderen Prozeß verwendet wird."              | <ul style="list-style-type: none"> <li>• Versuch, mit DEL eine Datei zu löschen, die gerade geöffnet / in Bearbeitung ist (</li> <li>• Versuch, mit RD das aktuelle Verzeichnis zu löschen</li> <li>• Versuch, mit RD ein Verzeichnis zu löschen, auf das noch eine Anwendung zugreift und das damit blockiert ist (z.B. unter WinWord, wenn das betreffende Verzeichnis gerade unter <i>Datei - Öffnen</i> ausgewählt wurde) !</li> </ul> |
| Fehlermeldung "Es sind keine weiteren Dateien vorhanden."   | <ul style="list-style-type: none"> <li>• erscheint beim Umbenennen eines Verzeichnisses mit REN, wenn das Verzeichnis das Attribut „s“ (Systemdatei) oder „h“ (Versteckte Datei) besitzt</li> </ul>  |
| Fehlermeldung "ECHO ist ausgeschaltet (OFF)" bzw. "ECHO ist eingeschaltet (ON)"   | Diese Meldung ist eigentlich keine Fehlermeldung und erscheint, wenn der ECHO-Befehl ohne Parameter aufgerufen wird.<br><u>Mögliche Fehlerursache:</u> ECHO-Befehl soll eigentlich den Inhalt einer Variablen ausgeben, die aber nicht definiert wurde oder leer ist.  |
| Fehlermeldung "Syntaxfehler" oder "... ist syntaktisch an dieser Stelle nicht verarbeitbar."                                | kann die verschiedensten Ursachen haben - z.B. wenn ein Befehl oder ein bestimmtes Schlüsselwort (z.B. "DO" beim FOR-Befehl) falsch geschrieben oder vergessen wurde - meist hilft ein Blick in die Dokumentation des betreffenden Befehls   |
| Fehlermeldung "Die Batch-Datei kann nicht gefunden werden."   | Batchdatei oder die Zugriffsrechte darauf wurden vermutlich zwischenzeitlich gelöscht (evtl. DEL- / RD-Befehl innerhalb der Batchdatei verantwortlich ?)   |
| Fehlermeldung "Die Datei kann nicht in sich selbst kopiert werden." bzw. "Datei kann nicht auf sich selbst kopiert werden." | z.B. wenn bei COPY / XCOPY der Name der Quell- und Zieldatei identisch ist   |
| Fehlermeldung "Ein Unterverzeichnis oder eine Datei mit dem Namen ... existiert bereits."                                   | <ul style="list-style-type: none"> <li>• Versuch, mit MD ein bereits existierendes Verzeichnis anzulegen bzw. ein Verzeichnis mit dem gleichen Namen wie der einer bereits existierenden Datei</li> </ul>  |
| Batchdatei wird unerwartet - offensichtlich ohne Fehlermeldung - beendet (Batchabbruch)                                     | <u>Mögliche Ursachen:</u><br><br>1. Versehentlich Umleitungssymbol "<" oder ">" in ECHO-Befehl verwendet, z.B. echo <Kapitel 1> (ggf. Escape-Symbol "^" voranstellen - siehe auch Kapitel "Befehlssymbole (Sonderzeichen mit speziellen Funktionen)")  |

|   |   |
|---|---|
|   | <p>2. Fehlermeldung "Das Sprungziel - xxx wurde nicht gefunden." erscheint, ist jedoch nicht lesbar, da die DOS-Box sofort wieder geschlossen wird</p> <ul style="list-style-type: none"> <li>• Batchdatei in bereits geöffneter DOS-Box (statt über Explorer) testen</li> <li>• in der Batchdatei verwendete Sprungmarken überprüfen</li> </ul>  |
| <p>Die folgende Verwendung des Pfadoperators zur Ersetzung eines Batchparameters ist ungültig: (z.B.: %~dpn)</p>  | <p>Syntaxfehler - hier fehlt z.B. die Nummer des zu verwendenden Batchparameters: %~dpn0 oder %~dpn1 etc. (Details - siehe Kapitel "Verwenden von Batchparametern")</p>   |
| <p>Fehlermeldung "CMD.EXE wurde mit dem oben angegebenen Pfad als aktuellem Verzeichnis gestartet.</p> <p><b>UNC-Pfade</b> werden nicht unterstützt. Stattdessen wird das Windows-Verzeichnis als aktuelles Verzeichnis gesetzt."</p> | <p><u>Ursache:</u></p> <ul style="list-style-type: none"> <li>• Batchdatei wurde über UNC-Pfad aufgerufen, z.B. über <i>Start - Ausführen</i> - \\Server\Freigabe\batchdatei.cmd<br/>Dies ist zwar durchaus möglich, allerdings unterstützt CMD.exe keinen UNC-Pfad als aktuelles Verzeichnis - beim Aufruf über UNC-Pfad wird daher die Fehlermeldung ausgegeben und das aktuelle Verzeichnis automatisch gewechselt</li> </ul>  |
| <p>Fehlermeldung "Das aktuelle Verzeichnis ist ungültig."</p>   | <p><u>Möglich Ursachen (siehe CD-Befehl):</u></p> <ul style="list-style-type: none"> <li>• das aktuelle Verzeichnis wurde umbenannt / verschoben / gelöscht</li> <li>• Zugriffsrechte des angemeldeten Benutzers auf das aktuelle Verzeichnis wurden gelöscht</li> <li>• aktuelles Verzeichnis befindet sich auf einem Server, der nicht mehr erreichbar ist</li> <li>• aktuelles Verzeichnis befindet sich auf einem Wechseldatenträger (Diskette, CD, etc.) der entfernt wurde</li> </ul> |
| <p>Fehlermeldung "Dateiname existiert bereits, oder die Datei konnte nicht gefunden werden."</p>  | <p><u>Ursache:</u></p> <p>Fehlerhafte Verwendung des Rename- (Ren-) Befehls:</p> <ul style="list-style-type: none"> <li>• Dateiname oder -pfad der Datei, die umbenannt werden soll, ist fehlerhaft</li> <li>• Neuer Dateiname, den die Datei erhalten soll, ist bereits vorhanden</li> </ul>   |
| <p>Fehlermeldung "Der Inhalt der Zieldatei wurde zerstört."</p>   | <p><u>Ursache:</u></p> <p>Beim Kopieren von mehreren Dateien in 1 Zieldatei ist eine der Quelldateien mit der Zieldatei identisch</p>   |

**Weitere Fehlermeldungen aus CMD.EXE (nur der Vollständigkeit halber - bisher nicht dokumentiert)**

| Batch-Fehler  | Mögliche Ursachen |
|---|-------------------|
| Fehlermeldung "Ein Prozeß hat versucht, zu einer nicht bestehenden Pipe zu schreiben."                      |                   |
| Fehlermeldung "Die eingegebene Zeile ist zu lang."  |                   |
| Fehlermeldung "Die Zugriffsnummer konnte nicht dupliziert werden, während der Zugriff umgeleitet wurde %1." |                   |
| Fehlermeldung "Die folgende Zeichenkette ist zu lang: ..."  |                   |
| Fehlermeldung "Die Zugriffsnummer konnte während eines Pipe-Vorgangs nicht dupliziert werden."              |                   |
| Fehlermeldung "Der Prozeß kann nicht abgeschlossen werden."   |                   |

## Regeln für gute Batchprogramme

### 1. "Routenplanung":

Vor Beginn der Programmierung überlegen, welche Schritte die Batchdatei erledigen soll + diese Schritte als Kommentare ins Programm schreiben (= nur aus Kommentaren bestehendes Grundgerüst des Programms).

Dann erst mit der Umsetzung der einzelnen Schritte in Programmbefehle beginnen.

### 2. "Wegfahrsperre":

Nach Aufruf einer Batchdatei zunächst Rückfrage an den Benutzer "Sind Sie sicher ... ?" oder entsprechende Funktionsbeschreibung des Programms ausgeben.

Dies ist besonders wichtig bei Batchdateien, die Installationen durchführen, Dateien löschen, Registry-Eingriffe vornehmen, etc. !

### 3. "Beschriftungen auf dem Armaturenbrett":

Kommentarzeilen verwenden ...

- um einem fachkundigen Anwender (= Batchprogrammierer) die Funktionsweise der Batchdatei bzw. einzelner Zeilen mit komplizierten Befehlen zu erläutern  
(aber nicht jeden Befehle erläutern: "xcopy ... " - REM Kopiert die Datei xxx nach yyy !)
- als Überschriften für die wichtigsten Schritte des Programms
- um die Voraussetzungen für die Batchdatei aufzulisten (z.B. "Verzeichnis xy muss bereits vorhanden sein und der Benutzer Schreibrechte darauf haben" oder "Batchdatei läuft nur auf PCs ohne CD-Laufwerk oder mehreren Partitionen")
- für Besonderheiten und Spezial-Tricks, die nicht von jedem Batchprogrammierer sofort zu durchschauen sind

### Empfehlung:

- Eingerückte Kommentarzeilen verwenden, wenn diese sich auf den Befehl unmittelbar in der Zeile davor beziehen
- Nicht eingerückte Kommentarzeilen als Überschriften verwenden, um die darauffolgenden Befehlszeilen zu erläutern

#### 4. Geeigneter "Parkplatz":

Batchdateien sinnvoll ablegen, z.B. eigenes Server-Verzeichnis für Tools und Batchdateien + Pfad auf den PCs um dieses Verzeichnis erweitern

#### 5. "Nur Zweitschlüssel verleihen":

Für Batchdateien, die im Netz o.ä. im Effektiv-Einsatz eigenes Verzeichnis mit den Originalen dieser Dateien anlegen, auf das nur die Systemverwaltung Zugriff hat.

Änderungen nur an dem Original durchführen und erst nach erfolgreichen Tests usw. in das Verzeichnis kopieren, in dem sich die Kopie befindet, die von den Benutzern im Effektiv-Betrieb aufgerufen wird.

#### 6. "Bedienungsanleitung":

Für größere + aufwendigere Batchdateien eigene Dokumentation über benötigte Verzeichnisse / Dateien / Zugriffsrechte, sonstige Voraussetzungen (Registry-Einträge, INI-Einträge, etc.), Überblick über die Funktionsweise des Programms, Von wem eingesetzt ? Welche Daten werden wohin transportiert ? usw.

## Weitere Informationen / Links zur Batchprogrammierung



Stand: 09.10.06