

HOWTO zu Shell-Exit-Codes

(C) 2006-2016 T.Birnthaler/H.Gottschalk <howtos(at)ostc.de>  
OSTC Open Source Training and Consulting GmbH  
<http://www.ostc.de>

\$Id: shell-exit-code-HOWTO.txt,v 1.9 2019/11/26 19:37:07 tsbirn Exp \$

Dieses Dokument beschreibt die Bedeutung und den Einsatz von Shell-Exit-Codes.

Jedes Kommando (jeder Prozess) gibt bei seinem Ende einen "Exit-Code" (kleiner Wert im Bereich 0-255) zurück, über den es seinem Vaterprozess mitteilt, ob es korrekt ablief oder ob ein Fehler auftrat.

Code	Bedeutung
0	Kommando lief korrekt ab
ungleich 0	Bei der Ausführung des Kommandos trat ein Fehler auf (Wert kann die Fehlerart beschreiben, muss aber nicht)

Die Bedeutung der einzelnen Exit-Fehlercodes ist von Kommando zu Kommando verschieden und kann in den jeweiligen man-Pages nachgelesen werden.

Der Exit-Code kann NUR direkt nach einem Kommando von Shell-Kontrollstrukturen abgefragt und über das Kommando "test" für Verzweigungen des Programmflusses benutzt werden. Er steht auch in der Shell-Variablen "\$?" (bzw. "\$status" bei der (t)csh) und kann darüber ausgegeben werden. Beispiele:

```
grep TEXT FILE      # 0 falls TEXT (Muster) in FILE gefunden
                   # 1 falls TEXT (Muster) in FILE nicht gefunden
                   # 2 falls FILE nicht vorhanden oder TEXT (Muster) fehlerhaft
echo $?             # Exit-Code ausgeben (nur 1x sinnvoll, da "echo" auch Kmdo.)
echo $?            # Exit-Code von vorherigem "echo" (Wert "0" da erfolgreich)
```

Da für die Shell der Wert 0 "logisch wahr" bedeutet und alle anderen Werte 1, 2, ... für sie "logisch falsch" bedeuten, kann der Exit-Code den Ablauf eines Shell-Skripts beeinflussen.

```
[ -f /tmp/text ] && echo "Datei '/tmp/text' existiert"
[ -d /tmp/dir ] || echo "Verzeichnis '/tmp/dir' existiert nicht"
```

Da jedes Kommando und jedes Programm einen Exit-Code erzeugt, kann sein Erfolg anschließend sofort überprüft und entsprechend reagiert werden (die Fehlermeldungen dabei auf "stderr" (1>&2) ausgeben):

```
grep "error:" /tmp/log > /dev/null && echo "Fehler in Log gefunden" 1>&2
mv text /tmp || echo "Datei 'text' nicht verschiebbar" 1>&2
chmod a+w log || echo "Rechte von Datei 'log' nicht änderbar" 1>&2

if grep "error:" /tmp/log > /dev/null
then
  echo "$(grep -c "error:" /tmp/log > /dev/null) Fehler in Log gefunden"
else
  echo "Alles in Ordnung"
fi
```

Auch die Shell selbst kennt einige fest vordefinierte Exit-Codes (bash, ksh):

Code	Bedeutung
126	Kommando gefunden, aber nicht lesbar/ausführbar
127	Kommando nicht gefunden
128 + NN	Signal NN führte zum Kommando-Abbruch