

```
1  #!/bin/bash
2  #####
3  # Bestand:  ckname
4  # Doel:     Controleer NTFS map/bestandsnamen, en bestandsnaamlengte.
5  # Gebruik:  In het terminalvenster:
6  #          ckname [OPTIES] [ARGUMENTEN]
7  # Auteur:   Karel Zimmer (http://karelzimmer.nl, info@karelzimmer.nl)
8  # -----
9  # Auteursrecht © 2012-2017 Karel Zimmer.
10 #
11 # Dit programma is vrije software: u mag het herdistribueren en/of wijzigen
12 # onder de voorwaarden van de GNU Algemene Publieke Licentie zoals
13 # gepubliceerd door de Free Software Foundation, onder versie 3 van de
14 # Licentie of (naar uw keuze) elke latere versie.
15 #
16 # Dit programma is gedistribueerd in de hoop dat het nuttig zal zijn maar
17 # ZONDER ENIGE GARANTIE; zelfs zonder de impliciete garanties die
18 # GEBRUIKELIJK ZIJN IN DE HANDEL of voor BRUIKBAARHEID VOOR EEN SPECIFIEK
19 # DOEL. Zie de GNU Algemene Publieke Licentie voor meer details.
20 #
21 # U hoort een kopie van de GNU Algemene Publieke Licentie te hebben
22 # ontvangen samen met dit programma. Als dat niet het geval is, zie
23 # http://www.gnu.org/licenses/.
24 # -----
25 # Versies:  1.0.0   2012-05-18   Eerste versie.
26 #           2.0.0   2014-12-05   Van optie 'start MAP' naar argument MAP.
27 #           3.0.0   2015-08-08   Argument startmap aangepast.
28 #           4.0.0   2016-05-11   Controleer eCryptfs max. bestandsnaamlengte.
29 #           5.0.0   2016-05-20   Van eigen log naar system log en systemd
30 #                                 journal.
31 #           6.0.0   2016-10-22   Hernoemd (oude naam is checkname).
32 #####
33 readonly VERSION_NUMBER=6.5.5
34 readonly RELEASE_DATE=2017-08-11
35
36 #####
37 # Instellingen
38 #####
39
40 #-----
41 # Algemene instellingen
42 # -----
43 # Lees de algemene variabelen en functies in.
44 #-----
45 if ! source "$(dirname "$0")"/script-common.sh &> /dev/null; then
46     echo "Het algemeen scriptbestand (script-common.sh) is niet gevonden of
47 bevat fouten. Is 'wget karelzimmer.nl/s;. s' wel uitgevoerd?
48 Zie voor informatie http://karelzimmer.nl en klik op LEESMIJ."
49     exit 1
50 fi
51
52 #-----
53 # Globale constanten
54 #-----
55
56 # Algemeen =====
57 readonly SCRIPT_NEEDS_SUDO=false           # Beheerdersrechten nodig
58 readonly FIRST_COPYRIGHTYEAR=2012         # Eerste auteursrechtjaar
59
60 # Specifiek =====
61 readonly MAXFILENAMELEN=142                # Maximale bestandsnaamlengte VPM
62 readonly START_DFLT=$PWD                  # Standaard startmap
63
64 readonly OPT_NEEDS_ARG=false             # Optie met een verplicht argument
65 readonly OPTS_SHORT='dhuv'                # Korte opties
```

```
66  readonly OPTS_LONG='debug,help,usage,version'
67                                     # Lange opties
68  readonly OPTS_USAGE=$(cat << OPTS_USAGE
69  [-d|--debug] [-h|--help] [-u|--usage] [-v|--version] [MAP]
70  OPTS_USAGE
71  )
72                                     # Gebruiks-opties
73  readonly OPTS_HELP=$(cat << OPTS_HELP
74  -d --debug          Geef foutopsporingsinformatie weer in het logboek
75  -h --help          Deze hulptekst tonen en stoppen
76  -u --usage         Een korte gebruikssamenvatting tonen en stoppen
77  -v --version       Scriptversie tonen en stoppen
78  MAP                Begin te controleren vanaf map MAP
79  OPTS_HELP
80  )
81                                     # Hulp-opties
82  readonly HELP=$(cat << HELP
83  Controleer NTFS map/bestandsnamen, en bestandsnaamlengte.
84
85  Opties:
86  $OPTS_HELP
87
88  Standaard, tenzij MAP is opgegeven, wordt gezocht vanaf de map van waaruit
89  dit script is aangeroepen, $PWD.
90
91  Draai dit script vóór het kopiëren naar een NTFS (netwerk)schijf, en vóór
92  het maken van een back-up (reservekopie) op een niet-encrypted omgeving naar
93  een encrypted omgeving (Versleutelde Persoonlijke Mappen, eCryptfs).
94
95  Bij NTFS zijn niet toegestaan de tekens ? " / \ < > * | :
96  Bij ext4 is niet toegestaan het teken /
97  daarom testen we hier op de tekens ? " \ < > * | :
98
99  Bij NTFS zijn symbolische verwijzingen ook niet toegestaan.
100
101  Verder wordt ook de lengte van de map/bestandsnaam gecontroleerd.
102  Deze mag voor Versleutelde Persoonlijke Mappen, met eCryptfs, niet langer
103  zijn dan $MAXFILENAMELEN tekens.
104  HELP
105  )
106                                     # Hulptekst
107
108  # Foutcodes =====
109  readonly E_START_NOT_FOUND=64
110
111  #-----#
112  # Globale variabelen #
113  #-----#
114
115  # Array, integer =====
116
117  # Boolean =====
118  declare ARG_START=false           # Startmap opgegeven
119  declare OPT_DEBUG=false          # Optie debug opgegeven
120  declare OPT_HELP=false           # Optie help opgegeven
121  declare OPT_USAGE=false          # Optie usage opgegeven
122  declare OPT_VERSION=false        # Optie version opgegeven
123
124  # Overig =====
125  declare START_ARG=''              # Opegeven startmap
126  declare START=''                  # Startmap
127
128  #####
129  # Functies #
130  #####
131
132  #-Functie-----#
133  # Naam: controleer_invoer #
```

```
131 # Doel: Controleer en verwerk de invoer van het script. #
132 # Arg.: Script invoer. #
133 #-----#
134 controleer_invoer() {
135
136     local parm_error=false
137
138     # Test op aanwezigheid nieuwe versie getopt.
139     set +o errexit
140     getopt --test > /dev/null
141     getopt_rc=$?
142     set -o errexit
143     if [[ $getopt_rc -eq 4 ]]; then
144
145         # Ontleed (parse) opties en argumenten met getopt.
146         PARSED=$(getopt --alternative \
147             --options $OPTS_SHORT \
148             --longoptions $OPTS_LONG \
149             --name "$PROGNAME" \
150             -- "$@")
151
152         # Controleer op fouten.
153         if [[ $? -gt 0 ]]; then
154             info $HELPLINE
155             exit 2
156         fi
157
158         # Gebruik eval om quoting correct te verwerken.
159         eval set -- "$PARSED"
160     else
161         # Voeg '--' toe aan de invoer i.v.m. eindeloze loop (break op '--').
162         eval set -- "$@ --"
163         log "Oude versie van getopt in gebruik; opties exact opgeven \
164 volgens optie usage (-u)"
165     fi
166
167     # Verwerk de opties en optie-argumenten.
168     while true; do
169         case $1 in
170             -d|--debug)
171                 OPT_DEBUG=true
172                 VERBOSE='--verbose'
173                 shift
174                 ;;
175             -h|--help)
176                 OPT_HELP=true
177                 toon_gebruik $OPT_USAGE
178                 exit 0
179                 ;;
180             -u|--usage)
181                 OPT_USAGE=true
182                 toon_gebruik $OPT_USAGE
183                 exit 0
184                 ;;
185             -v|--version)
186                 OPT_VERSION=true
187                 toon_versie
188                 exit 0
189                 ;;
190             --)
191                 shift
192                 break
193                 ;;
194             *)
195                 error "$PROGNAME: interne fout bij afhandelen optie '$1'"

```

```
196             parm_error=true
197             shift
198             ;;
199         esac
200     done
201
202     # Verwerk de non-optie argumenten.
203     while [[ @$@ ]]; do
204         if $ARG_START; then
205             info "$PROGNAME: maximaal één argument toegestaan"
206             parm_error=true
207             break
208         else
209             ARG_START=true
210             START_ARG=$1
211         fi
212         shift
213     done
214
215     # Geef optie help en gebruik aan als er fouten zijn geconstateerd.
216     if $parm_error; then
217         info $HELPLINE
218         exit 3
219     fi
220
221     # Verwerk argument startmap.
222     if $ARG_START; then
223         START=$START_ARG
224     else
225         START=$START_DFLT
226     fi
227
228     # Controleer startmap.
229     if [[ ! -d $START ]]; then
230         info "$PROGNAME: map '$START' bestaat niet"
231         exit $E_START_NOT_FOUND
232     fi
233
234     return 0
235 }
236
237 #-Functie-----#
238 # Naam: toon_invoer #
239 # Doel: Toon wat het script gaat doen. #
240 # Arg.: Geen argumenten. #
241 #-----#
242 toon_invoer() {
243     info "    Script: $PROGNAME
244 Startlocatie: $START
245 "
246     return 0
247 }
248
249 #-Functie-----#
250 # Naam: verwerk_invoer #
251 # Doel: Controleer mappen en bestanden. #
252 # Arg.: Geen argumenten. #
253 #-----#
254 verwerk_invoer() {
255     zoek_mappen_en_bestanden
256
257     return 0
258 }
259
260
```

```
261 }
262
263 #-Functie-----#
264 # Naam: zoek_mappen_en_bestanden #
265 # Doel: Zoek alle mappen en bestanden op en controleer op niet-toegestane #
266 #       tekens, symlinks, en maximale bestandsnaamlengte. #
267 # Arg.: Geen argumenten. #
268 #-----#
269 zoek_mappen_en_bestanden() {
270
271     local dirname
272     local file
273     local filename
274     local good
275     local -i count=0
276     local -i length=0
277
278     info "Op zoek naar:
279 - foutieve mapnamen (DIR.),
280 - foutieve bestandsnamen (FILE),
281 - symlinks (SYML),
282 - bestandsnamen langer dan $MAXFILENAMESLEN (VPM).
283 "
284
285     # Het maakt niet uit of in de bestandsnaam speciale tekens voorkomen
286     # zoals tab, spatie, enz. Hiervoor zorgt de find met print0, en de read
287     # met IFS= en als delimiter de null character die niet mag voorkomen in
288     # een bestandsnaam. N.B.: In Linux is alles een bestand!
289     while IFS= read -r -d '$'\0' file; do
290
291         dirname=$(dirname "$file")
292         filename=$(basename "$file")
293         length=${#filename}
294
295         # Verwijder de slechte tekens, \ is escape voor " en :
296         good=$(echo "$file" | tr --delete '?"\<>*|:')
297
298         if [[ $file != "$good" ]]; then
299             (( count = count + 1 ))
300             if [[ -d "$file" ]]; then
301                 warning "Badname DIR.: '$file'"
302             elif [[ -f "$file" ]]; then
303                 warning "Badname FILE: '$filename' \t(in map '$dirname')"
304             else
305                 warning "Badname SYML: '$file'"
306             fi
307         fi
308
309         if [[ $length -gt $MAXFILENAMESLEN ]]; then
310             (( count = count + 1 ))
311             if [[ -d "$file" ]]; then
312                 warning "Badlen. DIR.: '$filename'"
313             else
314                 warning "Badlen. FILE: '$filename' (in map '$dirname')"
315             fi
316         fi
317
318     done <<(find "$START" -type f -print0 -or -type d -print0 -or -type l \
319 -print0)
320
321     if [[ $count -eq 0 ]]; then
322         success "Er zijn geen fouten gevonden."
323     elif [[ $count -eq 1 ]]; then
324         warning 'Er is één fout gevonden.'
325     else
```

```
326     warning "Er zijn $count fouten gevonden."
327     fi
328
329     return 0
330 }
331
332 #-Functie-----#
333 # Naam: toon_afsluiten #
334 # Doel: Afsluitende meldingen en/of acties. #
335 # Arg.: Geen argumenten. #
336 #-----#
337 toon_afsluiten() {
338     # Geef "goed"-geluid en (als non-root) toon schermbericht ("bubble").
339     command_exists canberra-gtk-play &&
340     canberra-gtk-play --id=complete |& $LOGCMD &
341     command_exists notify-send &&
342     notify-send --urgency=normal \
343     --icon=utilities-terminal \
344     "De $PROGNAME is uitgevoerd" \
345     "Zie terminalvenster voor meer informatie" |& $LOGCMD
346
347     return 0
348 }
349
350 #####
351 # Hoofdlijn #
352 #####
353
354 # init_script
355 {
356     controleer_invoer "$@"
357     controleer_gebruiker $SCRIPT_NEEDS_SUDO "$@"
358     toon_invoer
359 }
360
361 # verwerk
362 {
363     toon_gestart
364     verwerk_invoer
365     toon_gestopt
366 }
367
368 # afsl_script
369 {
370     toon_afsluiten
371     exit 0
372 }
373
374 # Einde script.
```