

```

1  #!/bin/bash
2  #####
3  # Bestand:  checkname.sh                                     #
4  # Doel:    Script voor het controleren van NTFS map- en bestandsnamen. #
5  # Gebruik: ./checkname.sh [opties]                           #
6  # Gebruikt: script-common.sh (algemene variabelen en functies) #
7  # Auteur:  Karel Zimmer (http://karelzimmer.nl, info@karelzimmer.nl) #
8  # -----#
9  # Auteursrecht © 2012-2014 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 #####
28 readonly VERSION_NUMBER=2.0.0
29 readonly RELEASE_DATE=2014-12-05
30
31 #####
32 # Instellingen                                                    #
33 #####
34
35 #-----#
36 # Algemene instellingen                                           #
37 # -----#
38 # Lees de algemene variabelen en functies in.                     #
39 #-----#
40 source script-common.sh 2> /dev/null || {
41     echo "Het algemeen scriptbestand 'script-common.sh'"          >&2
42     echo 'is niet gevonden of bevat fouten.'                      >&2
43     echo 'Dit algemeen scriptbestand wordt gewoonlijk door'
44     echo "script getscripts (gs) gedownload en in '/tmp' geplaatst."
45     echo "Is 'cd /tmp; wget karelzimmer.nl/gs; bash gs' uitgevoerd?"
46     echo 'Voor scripts zie http://karelzimmer.nl, klik op Linux.'
47     exit 1
48 }
49
50 #-----#
51 # Globale constanten                                              #
52 # -----#
53 # Algemeen -----#
54 readonly SCRIPT_NEEDS_SUDO=false # Uitvoeren als standaardgebruiker
55 readonly FIRST_COPYRIGHT_YEAR=2012 # Eerste auteursrechtjaar
56 readonly OPTIONS_HELP=$(cat << OPTIONS_HELP
57     MAP                begin te controleren vanaf map MAP
58 OPTIONS_HELP
59 ) # Extra hulp-opties (-h)
60 readonly OPTIONS_USAGE=$(cat << OPTIONS_USAGE
61
62     [MAP]
63 OPTIONS_USAGE
64 ) # Extra gebruiks-opties (-u)
65
66 # Specifiek -----#

```

```

67 # Foutcodes -----#
68 readonly E_STARTLOC_NOT_FOUND=64
69
70 #-----#
71 # Globale variabelen #
72 #-----#
73 # Normaal -----#
74 declare STARTLOC="$(pwd)" # Startlocatie
75
76 # Array, integer -----#
77 # Boolean -----#
78
79 #####
80 # Functies (op volgorde van uitvoering) #
81 #####
82
83 #-Functie-----#
84 # Naam: toon_hulp #
85 # Doel: Uitleg werking script. #
86 # Arg.: Geen argumenten. #
87 #-----#
88 toon_hulp() {
89     toon_gebruik "$SCRIPT_NEEDS_SUDO" "$OPTION_USAGE"
90     cat << HULP
91
92     Beschrijving:
93     Script voor het controleren van NTFS map- en bestandsnamen.
94
95     Dit script controleert op toegestane tekens in map- en bestandsnamen.
96     Handig om te draaien vóór het kopiëren naar een NTFS (netwerk)schijf.
97
98     Bij NTFS zijn niet toegestaan de tekens ? " / \ < > * | :
99     Bij ext4 is niet toegestaan het teken /
100     daarom testen we hier op de tekens ? " \ < > * | :
101     Bij NTFS zijn symbolische verwijzingen ook niet toegestaan.
102
103     Als er bij de aanroep geen MAP is opgegeven wordt gezocht vanaf de map
104     van waaruit dit script is aangeroepen.
105
106     $OPTIONS_HELP_SC
107     $OPTIONS_HELP
108     HULP
109 }
110
111 #-Functie-----#
112 # Naam: controleer_invoer #
113 # Doel: Initiële controles en/of acties. #
114 # Arg.: Geen argumenten. #
115 #-----#
116 controleer_invoer() {
117     if [[ ! -e $STARTLOC ]]; then
118         error "Map $STARTLOC bestaat niet."
119         exit $E_STARTLOC_NOT_FOUND
120     fi
121 }
122
123 #-Functie-----#
124 # Naam: toon_invoer #
125 # Doel: Toon wat het script gaat doen. #
126 # Arg.: Geen argumenten. #
127 #-----#
128 toon_invoer() {
129     clear
130     log "$DASHES"
131     normal "$HEADER"
132     normal

```

```

133     normal "Startlocatie: $STARTLOC"
134     normal "Logboek:      $LOG"
135     log    "$DASHES"
136     normal
137 }
138
139 #-Functie-----#
140 # Naam: zoek_mappen                                #
141 # Doel: Zoek alle mappen op en controleer op niet-toegestande tekens. #
142 # Arg.: Geen argumenten.                            #
143 #------#
144 zoek_mappen() {
145     local file
146     local good
147     local -i count=0
148
149     normal 'Op zoek naar foutieve mapnamen (DIR)...'
150
151     #-----#
152     # Het maakt niet uit of in de bestandsnaam speciale tekens voorkomen #
153     # zoals tab, spatie, enz. Hiervoor zorgt de find met print0,          #
154     # en de read met IFS= en als delimiter de null character die         #
155     # niet mag voorkomen in een bestandsnaam.                            #
156     #-----#
157     while IFS= read -r -d $'\0' file; do
158
159         # Verwijder de slechte tekens, \ is escape voor " en :
160         good=$(echo "$file" | tr --delete '\?\"\\<>\\*\\|\\:')
161
162         if [[ $file != $good ]]; then
163             let count++
164             normal "Badname DIR.: $file"
165         fi
166     done < < (
167         find "$STARTLOC" \
168             -type d \
169             -print0
170     )
171
172     if [[ $count -eq 0 ]]; then
173         success 'Er zijn geen foutieve mapnamen gevonden.'
174     elif [[ $count -eq 1 ]]; then
175         warning 'Er is één foutieve mapnaam gevonden.'
176     else
177         warning "Er zijn $count foutieve mapnamen gevonden."
178     fi
179 }
180
181 #-Functie-----#
182 # Naam: zoek_bestanden                                #
183 # Doel: Zoek alle bestanden op en controleer op niet-toegestande tekens. #
184 # Arg.: Geen argumenten.                            #
185 #------#
186 zoek_bestanden() {
187     local dirname
188     local file
189     local filename
190     local good
191     local -i count=0
192
193     normal 'Op zoek naar foutieve bestandsnamen (FILE) en symlinks (SYML)...'
194
195     while IFS= read -r -d $'\0' file; do
196
197         good=$(echo "$file" | tr --delete '\?\"\\<>\\*\\|\\:')
198

```

```

199         if [[ $file != $good ]]; then
200             let count++
201             filename=$(basename "$file")
202             dirname=$(dirname "$file")
203             normal "Badname FILE: $filename in $dirname"
204         fi
205
206         if [[ -h $file ]]; then
207             let count++
208             normal "Badname SYML: $file"
209         fi
210     done < < (
211         find "$STARTLOC" \
212             -type f \
213             -print0 \
214             -or \
215             -type l \
216             -print0
217     )
218
219     if [[ $count -eq 0 ]]; then
220         success 'Er zijn geen foutieve bestandsnamen en symlinks gevonden.'
221     elif [[ $count -eq 1 ]]; then
222         warning 'Er is één foutieve bestandsnaam en symlink gevonden.'
223     else
224         warning "Er zijn $count foutieve bestandsnamen en symlinks gevonden."
225     fi
226 }
227
228 #-Functie-----#
229 # Naam: toon_afsluiten #
230 # Doel: Afsluitende meldingen en/of acties. #
231 # Arg.: Geen argumenten. #
232 #-----#
233 toon_afsluiten() {
234     : # no-op
235 }
236
237 #####
238 # Hoofdlijn #
239 #####
240 # init_script
241 {
242     verwerk_invoer "$@"
243     controleer_gebruiker "$SCRIPT_NEEDS_SUDO"
244     bepaal_log "$SCRIPT_NEEDS_SUDO" LOG
245     controleer_invoer
246 }
247 # verwerk
248 {
249     toon_invoer
250     toon_gestart
251     zoek_mappen
252     zoek_bestanden
253     toon_gestopt
254 }
255 # afsl_script
256 {
257     toon_afsluiten
258     toon_afsluiten_sc
259     exit $MAXRC
260 }
261
262 # Einde script.

```