

```
1 #####
2 # Bestand:  script-common.sh                               #
3 # Doel:    Algemene variabelen en functies voor scripts.  #
4 # Gebruik: . script-common.sh [NAAM]                     #
5 #         -of-                                           #
6 #         source script-common.sh [NAAM]                 #
7 #                                                     #
8 #         Opties: NAAM  gebruik NAAM als naam (PROGNAME) voor <script> #
9 #                   als deze wordt gesourced (. <script> of #
10 #                   source <script>), anders is bash de naam #
11 #                   (PROGNAME) voor <script>             #
12 # Auteur:   Karel Zimmer (http://karelzimmer.nl, info@karelzimmer.nl) #
13 # -----#
14 # Een script dient dit bestand ("library file") in te kopiëren ("include"). #
15 # Dit om te voldoen aan de zogenaamde "SPOT rule" ("Single Point of Truth #
16 # regel"): alle scripts gebruiken dezelfde variabelen en functies.        #
17 # Zie script template, en anderen, voor meer informatie.                  #
18 # -----#
19 # Dit werk valt onder een Creative Commons Naamsvermelding-GelijkDelen 4.0 #
20 # Internationaal licentie (CC BY-SA 4.0).                                #
21 # Bezoek http://creativecommons.org/licenses/by-sa/4.0/deed.nl om een kopie #
22 # te zien van de licentie of stuur een brief naar Creative Commons,       #
23 # PO Box 1866, Mountain View, CA 94042, USA.                             #
24 # -----#
25 # Versies1: 1.0.0   2009-06-17  Eerste versie.                #
26 #             2.0.0   2009-11-12  Berichten in kleur.         #
27 #             3.0.0   2011-01-02  Voortgangsberichten.       #
28 #             4.0.0   2011-03-06  Versleutelde persoonlijke mappen (VPM). #
29 #             5.0.0   2011-11-08  Niet-versleutelde back-up/restore VPM. #
30 #             6.0.0   2012-04-22  Back-up/restore VPM als "cleartext data". #
31 #             7.0.0   2012-12-13  Vraag om [sudo] wachtwoord indien nodig. #
32 #             8.0.0   2014-10-31  Verwerk scriptargumenten zonder opties. #
33 #             9.0.0   2014-11-18  Eén logboekmap voor iedere type gebruiker. #
34 #            10.0.0   2014-12-07  Verwerk een niet-sudo gebruiker.        #
35 #            11.0.0   2015-07-26  Logboek naar dagelijks i.p.v. timestamped. #
36 #            12.0.0   2015-08-05  Logboek etc. naar ~/bin i.p.v. ~/.scripts #
37 #                                     en weer timestamped.          #
38 #            13.0.0   2015-10-10  Functie verwerk_rc van 2 naar 3 parameters. #
39 #            14.0.0   2016-05-20  Van eigen log naar system log en systemd #
40 #                                     journal.                       #
41 #            15.0.0   2017-03-20  Ontleden opties en argumenten door getopt. #
42 #            16.0.0   2017-06-01  Met optie debug opdrachtinformatie en #
43 #                                     scriptverloop naar de log.    #
44 #            17.0.0   2017-06-20  Script hardening met 'set -o' van errexit, #
45 #                                     nounset, en pipefail, en 'trap exit'. #
46 #                                     #                               #
47 # -----#
48 # 1 Toelichting versienummering en -datum:                    #
49 #   V.R.M   VER.-DATUM  Versie.Release.Modificatienummer en datum #
50 #   =.=.=   =====   Alleen commentaar of spatiëring gewijzigd. #
51 #   =.=.+   JJJJ-MM-DD  Kleine technische wijziging.             #
52 #   =.+0    JJJJ-MM-DD  Kleine functionele wijziging.           #
53 #   +.0.0   JJJJ-MM-DD  Nieuwe functionaliteit.                 #
54 #####
55 # VERSION_NUMBER=17.4.0          # Versienummer
56 # RELEASE_DATE=2017-08-17       # Versiedatum
57
58 #####
59 # Instellingen
60 #####
61 set -o errexit          # Stop als opdracht niet-nul eindigt
62 set -o nounset         # Stop als variabele ongebonden is
63 set -o pipefail       # Pipe eindigt met hoogste rc
64 #et -o noexec         # Opdrachten niet uitvoeren
65 #et -o verbose       # Opdracht tonen voor uitvoering
```

```

66 #et -o xtrace # Resultaat van de opdracht tonen
67
68 #-----#
69 # Algemene globale constanten #
70 #-----#
71
72 # Algemeen =====
73 readonly ASK_FOR_PW_HELPTEXT="Indien nodig wordt het beheerderswachtwoord \
74 gevraagd." # Vraag om beheerderswachtwoord
75 readonly BINDIR=$HOME/bin # Direct-uitvoerbare-scriptsmap
76 readonly DASHES=$(printf '\u002D%.0s' {1..77})
77 # Streepregel, Unicode002D='-'
78 readonly DEVMACH='PC06' # Ontwikkelmachine
79 readonly DEVUSER='karel' # Ontwikkelaar
80 readonly PART_OF_INSTALL_HELPTEXT=$(cat << PART_OF_INSTALL_HELPTEXT
81 Dit script wordt ook gebruikt bij het installeren van Linux.
82 Ga voor meer informatie naar de site http://karelzimmer.nl en
83 kijk naar Checklist installatie.
84 PART_OF_INSTALL_HELPTEXT
85 ) # Installatietekst
86 # Bij ssh-connectie kan $0 '-bash' opleveren, vandaar na -- als argument.
87 readonly PROG=$(readlink --canonicalize -- "$0")
88 # Absoluut script (bijvoorbeeld
89 # geen './gset' maar '/tmp/gset')
90 readonly PROGDIR_ORG=$(dirname "$PROG") # Scriptmap oorspronkelijk
91 readonly PROGNAME_ORG=$(basename "$PROG")
92 # Scriptnaam oorspronkelijk
93 declare PROGDIR=$PROGDIR_ORG # Scriptmap standaard
94 [[ $PROGNAME_ORG = *bash ]] && PROGDIR=$(pwd)
95 # Scriptmap als sourced
96 readonly PROGDIR # Scriptmap
97 declare PROGNAME=$PROGNAME_ORG # Scriptnaam standaard
98 [[ $PROGNAME_ORG = *bash ]] && PROGNAME=${1:-$PROGNAME_ORG}
99 # Scriptnaam als sourced
100 readonly PROGNAME # Scriptnaam
101 readonly DISTID=$(lsb_release --id --short)
102 # Distributie-ID, bijvoorbeeld Ubuntu
103 [[ $DISTID = 'Debian' ]] && readonly DEBIAN=true ||
104 readonly DEBIAN=false # Debian-indicator
105 [[ $DISTID = 'Ubuntu' ]] && readonly UBUNTU=true ||
106 readonly UBUNTU=false # Ubuntu-indicator
107 [[ -d /usr/share/lubuntu ]] && readonly LUBUNTU=true ||
108 readonly LUBUNTU=false # Lubuntu-indicator
109 [[ -d /usr/share/xubuntu ]] && readonly XUBUNTU=true ||
110 readonly XUBUNTU=false # Xubuntu-indicator
111 readonly DISTREL=$(lsb_release --release --short)
112 # Distributieverie, bijv. 12.04
113 echo ${HOSTNAME:0:2} | grep --ignore-case --quiet 'SV' &&
114 readonly EDITION='server' ||
115 readonly EDITION='desktop' # Editie
116 readonly HELPLINE="Typ '$PROGNAME --help' of '$PROGNAME --usage' voor meer \
117 informatie." # Hulpregel
118 readonly LESS_OPTIONS='-FRSXMK' # Less-opties, overgenomen van
119 # systemctl en journalctl, zie bijv.
120 # 'man systemctl', zoek $SYSTEMD_LESS
121 command -v systemd &> /dev/null &&
122 readonly LOGCMD="systemd-cat --identifier=$PROGNAME --priority=info" ||
123 readonly LOGCMD="logger --tag $PROGNAME --priority info"
124 # Logging-opdracht
125 # ⓘ Toelichting system log en systemd journal:
126 # - Logging naar systemd journal op systemd-systemen met:
127 # <opdracht> |& systemd-cat -t $PROGNAME -p info
128 # Info uit de journal met:
129 # journalctl -ab -t $PROGNAME [-S <tijd>]
130 # - Logging naar system log op non-systemd-systemen met:

```

```

131 # <opdracht> |& logger -t $PROGNAME -p info
132 # Info uit de log met:
133 # grep $PROGNAME /var/log/syslog
134 readonly SPACES=$(printf '%.0s' {1..77})
135 # Spatieregel, ' '
136 readonly SUDO_PROMPT=$(cat << SUDO_PROMPT
137 Script $PROGNAME moet worden uitgevoerd met beheerdersrechten.
138 sudo $PROGDIR/$PROGNAME @$
139 [sudo] wachtwoord voor %p:
140 SUDO_PROMPT
141 ) # Prompt als beheerder
142 readonly TIMESTAMP=$(date +%Y-%m-%d-%H.%M.%S.%3N)
143 # Tijdstempel JJJJ-MM-DD-UU.MM.SS.MMM
144 # Lees gebruikersmapnamen in:
145 if ! source ${XDG_CONFIG_HOME:-$HOME/.config}/user-dirs.dirs && /dev/null
146 then
147     if locale | grep --quiet '=nl'; then
148         readonly XDG_DESKTOP_DIR=$HOME/Bureaublad
149         readonly XDG_DOWNLOAD_DIR=$HOME/Downloads
150         readonly XDG_TEMPLATES_DIR=$HOME/Sjablonen
151         readonly XDG_PUBLICSHARE_DIR=$HOME/Openbaar
152         readonly XDG_DOCUMENTS_DIR=$HOME/Documenten
153         readonly XDG_MUSIC_DIR=$HOME/Muziek
154         readonly XDG_PICTURES_DIR=$HOME/Afbeeldingen
155         readonly XDG_VIDEOS_DIR="$HOME/Video's"
156     else
157         readonly XDG_DESKTOP_DIR=$HOME/Desktop
158         readonly XDG_DOWNLOAD_DIR=$HOME/Downloads
159         readonly XDG_TEMPLATES_DIR=$HOME/Templates
160         readonly XDG_PUBLICSHARE_DIR=$HOME/Public
161         readonly XDG_DOCUMENTS_DIR=$HOME/Documents
162         readonly XDG_MUSIC_DIR=$HOME/Music
163         readonly XDG_PICTURES_DIR=$HOME/Pictures
164         readonly XDG_VIDEOS_DIR=$HOME/Videos
165     fi
166 fi
167
168 # Tekstattributen, zie ook: man terminfo. Tekst <opmaak> <functie>
169 readonly N=$(tput sgr0) # Tekst normaal info
170 readonly B=$(tput bold) # Tekst vet info --bold
171 readonly C=${B}${tput setaf 4} # Tekst blauw info --command
172 readonly I=$(tput sitm) # Tekst schuin
173 readonly S=${B}${tput setaf 2} # Tekst groen success
174 tput longname | grep --quiet 256 &&
175 readonly W=${B}${tput setaf 208} || # Tekst oranje warning
176 readonly W=${B}${tput setaf 3} # Tekst geel warning
177 readonly E=${B}${tput setaf 1} # Tekst rood error
178 # Voorbeeld kleurgebruik:
179 # Altijd ${<opmaak>} zodat het duidelijk is dat het opmaak betreft.
180 # - echo -e "${E}Foutje${N}"
181 # - echo -en {E}; echo 'Foutje'; echo -en ${N}
182 # (voordeel is dat dan er geen stuurtekens zichtbaar zijn)
183 # - voorbeeld gebruik functie: error 'Foutje'
184
185 # Diverse statussen:
186 readonly STATUS_SIMULATE="[${W}SIMULATIE${N}]"
187 readonly STATUS_SUCCESS="[${S}GOED${N}]"
188 readonly STATUS_ERROR="[${E}FOUT${N}]"
189
190 # Specifiek =====
191 readonly LASTRUNDIR=$HOME/.lastrun # Laatste-uitvoeringsmap
192 readonly LASTRUN=$LASTRUNDIR/lastrun-$PROGNAME
193 # Laatste-uitvoeringsbestand
194 readonly SUPERUID=0 # UserID beheerder, superuser, root
195 readonly THIS_YEAR=$(date +%Y) # Dit jaar (JJJJ)

```

```
196
197 # Foutcodes =====
198 readonly E_SUDO_NOT_ALLOWED=100
199 readonly E_WE_ARE_SUPERUSER=101
200 readonly E_SIGNAL_EXIT=102
201 readonly E_COMMAND_NOT_EXECUTABLE=126
202 readonly E_COMMAND_NOT_FOUND=127
203 readonly E_INVALID_EXIT_ARGUMENT=128
204 readonly E_SIGPIPE=141
205 readonly E_INVALID_EXIT_STATUS=255
206
207 #-----#
208 # Algemene globale variabelen #
209 #-----#
210
211 # Array, integer =====
212
213 # Boolean =====
214 declare OPT_DEBUG=false # Optie debug opgegeven
215 declare OPT_HELP=false # Optie help opgegeven
216 declare OPT_STARTER=false # Optie starter (in desktopbestand)
217 declare OPT_USAGE=false # Optie usage opgegeven
218 declare OPT_VERSION=false # Optie version opgegeven
219
220 # Overig =====
221 declare PARSED='' # Door getopt ontlede opties/argum.
222 declare SPINNER_DELAY=0.1 # Tijd tussen spinafbeeldingen
223 declare SPINNER_PID='' # Spinner-proces-ID
224 declare STATIME=0 # Timer's starttijd
225 declare STOTIME=0 # Timer's eindtijd
226 declare TO_DELETE='' # Evt. te verwijderen back-up
227 declare VERBOSE='' # Meer opdrachtinformatie
228
229 #####
230 # Algemene functies (op alfabetische volgorde) #
231 #####
232
233 #-Functie-----#
234 # Naam: attendeer_gebruiker #
235 # Doel: Geef "attentie"-geluid en optioneel vraag gebruiker om reactie. #
236 # Arg.: Eén optioneel argument: #
237 # 1. tekst string, invoer #
238 #-----#
239 attendeer_gebruiker() {
240
241     canberra-gtk-play --id=message \
242                     --loop=3      |& $LOGCMD || true
243
244     if [[ $@ ]]; then
245         read -p "$@"
246     fi
247
248     return 0
249 }
250
251
252 #-Functie-----#
253 # Naam: command_exists #
254 # Doel: Controleer of opdracht beschikbaar is op het systeem. #
255 # Arg.: Geen argumenten #
256 # Afsluitwaarde: 0 opdracht is beschikbaar #
257 #                1 opdracht is niet beschikbaar #
258 #-----#
259 command_exists() {
260
```

```
261     command -v "$@" &> /dev/null
262
263 }
264
265 #-Functie-----#
266 # Naam: controleer_gebruiker #
267 # Doel: Bekijkt of de gebruiker de juiste rechten heeft, traps zetten, en #
268 #       debug-sessie starten. #
269 # Arg.: Twee verplichte argumenten: #
270 #       1. script_needs_sudo   boolean, invoer: #
271 #           true   we moeten beheerder (superuser of root) zijn #
272 #           false  we moeten standaardgebruiker zijn #
273 #       2. Script-opties en -argumenten. #
274 #-----#
275 controleer_gebruiker() {
276
277     local script_needs_sudo=${1:-false}
278     local grp=''
279
280     # We moeten root zijn, controleer of dat zo is.
281     if $script_needs_sudo && [[ $UID -ne $SUPERUID ]]; then
282
283         # Maak een lijst van toegestane (en verboden) opdrachten voor de
284         # gebruiker. Als dit goed gaat mag de gebruiker sudo gebruiken en de
285         # lijst niet tonen. Als dit fout gaat mag de gebruiker geen sudo
286         # gebruiken, toon foutmelding, geef meer informatie en toon
287         # gebruikers.
288         sudo --list \
289             --prompt \
290             "$SUDO_PROMPT" | $LOGCMD
291         if [[ $? -gt 0 ]]; then
292             info "Om script $PROGNAME te kunnen uitvoeren zijn \
293 beheerdersrechten nodig,
294 met 'sudo', of als root.
295 Mogelijke gebruikers met beheerdersrechten zijn:
296 $(
297     while read record; do
298         grp=$(echo "$record" | awk -F ':' '{print $4}')
299         if [[ $grp -ge 1000 && $grp -ne 65534 ]]; then
300             echo "$record" | awk -F ':' '{print $1}'
301         fi
302     done < /etc/passwd
303     )
304 "
305             if $OPT_STARTER; then
306                 if [[ $PROGNAME = 'backup' ]]; then
307                     info "Gebruik starter 'Backup --no-root uitvoeren' om \
308 alleen de eigen Persoonlijke map veilig te stellen."
309                 elif [[ $PROGNAME = 'restore' ]]; then
310                     info "Gebruik starter 'Restore --no-root uitvoeren' om \
311 alleen de eigen Persoonlijke map terug te zetten."
312                 fi
313             else
314                 info "Typ bijvoorbeeld 'su beheerder', waarbij beheerder \
315 voldoende rechten heeft,
316 of meld u aan als beheerder, en voer daarna script \
317 $PROGNAME opnieuw uit."
318                 fi
319                 exit $E_SUDO_NOT_ALLOWED
320             fi
321
322             # Start het script opnieuw, maar dan met 'sudo'.
323             shift # Het eerste argument (script_needs_sudo) is verwerkt
324             log "Start script $PROGDIR/$PROGNAME opnieuw met sudo..."
325             sudo --preserve-env $PROGDIR/$PROGNAME "$@"

```

```

326     exit $?
327 fi
328
329 # We moeten niet-root zijn, controleer of dat zo is.
330 if ! $script_needs_sudo && [[ $UID -eq $SUPERUID ]]; then
331     error "Script $PROGNAME mag niet met beheerdersrechten (met 'sudo', \
332 of als root) worden uitgevoerd."
333     exit $E_WE_ARE_SUPERUSER
334 fi
335
336 shift # Het eerste argument (script_needs_sudo) is verwerkt
337
338 # Trap pas hier aanzetten, anders dubbele "trap signal" bij een root-
339 # script die gestart wordt als een standaardgebruiker i.p.v. als een
340 # beheerder.
341 trap verwerk_exit EXIT
342 trap "verwerk_signal INT" SIGINT
343 trap "verwerk_signal TERM" SIGTERM
344 trap "verwerk_signal HUP" SIGHUP
345
346 # Fifo (named-pipe) pas hier, anders dubbele .xtrace-bestanden (met
347 # verschillende eigenaren) bij een root-script die gestart wordt als een
348 # standaardgebruiker i.p.v. als een beheerder.
349 if $OPT_DEBUG; then
350     warning "\
351 #####\n\
352 # START DEBUG-SESSIE #\n\
353 #####"
354     # Met fifo (named pipe) de xtrace-output naar syslog/journal.
355     # Maak fifo (named pipe).
356     PIPE=$(mktemp -t $PROGNAME-XXXXXXX.xtrace)
357     rm $VERBOSE --force $PIPE |& $LOGCMD
358     mkfifo $PIPE
359     # Redirect fifo (named pipe) naar syslog/journal.
360     $LOGCMD < $PIPE &
361     # Open file descriptor 4 naar fifo (named pipe).
362     exec 4>$PIPE
363     # De xtrace-output gaat (ook) naar file descriptor 4.
364     BASH_XTRACEFD=4
365     set -o xtrace
366     # PS4-prompt met extra info.
367     ps4='+(${BASH_SOURCE}:${LINENO}): ${FUNCNAME[0]:+${FUNCNAME[0]}(): }'
368     export PS4=$ps4
369     unset ps4
370 fi
371
372 return 0
373 }
374
375 #-Functie-----#
376 # Naam: error #
377 # Doel: Tekst rood tonen naar stderr. #
378 # Arg.: Eén optioneel argument: #
379 # 1. tekst string, invoer #
380 #-----#
381 error() {
382
383     echo -e "${E}$@${N}" >&2
384
385     return 0
386 }
387
388 #-Functie-----#
389 # Naam: info #
390 # Doel: Tekst tonen, eventueel vet (extra opvallend) of blauw (voor #

```

```
391 #         opdrachten).                                     #
392 # Arg.: Twee optionele argumenten:                         #
393 #         1. optie     string, invoer:                     #
394 #         -b, --bold, -c, --command                       #
395 #         2. tekst     string, invoer                     #
396 #-----#
397 info() {
398
399     local optie=${N}
400     local argum=''
401
402     while [[ @$@ ]]; do
403         case $1 in
404             -b|--bold)
405                 optie=${B}
406                 shift
407                 continue
408                 ;;
409             -c|--command)
410                 optie=${C}
411                 shift
412                 continue
413                 ;;
414             *) # Argument tekst.
415                 if [[ -z $argum ]]; then
416                     argum=$1
417                 else
418                     argum="$argum $1"
419                 fi
420                 shift
421                 continue
422                 ;;
423         esac
424     done
425
426     echo -e "${optie}$argum${N}"
427
428     return 0
429 }
430
431 #-Functie-----#
432 # Naam: log                                             #
433 # Doel: Tekst naar de log/journal sturen bij optie debug. #
434 # Arg.: Eén optioneel argument:                       #
435 #         1. tekst     string, invoer                     #
436 #-----#
437 log() {
438
439     $OPT_DEBUG && echo -e "$@" |& $LOGCMD
440
441     return 0
442 }
443
444 #-Functie-----#
445 # Naam: spinner                                         #
446 # Doel: Laat een spinner verschijnen of verdwijnen.    #
447 # Arg.: Eén verplicht argument:                       #
448 #         1. spinoptie string, invoer:                 #
449 #         aan         spinner verschijnt en draait, of #
450 #         uit         spinner stopt en verdwijnt.      #
451 #-----#
452 spinner() {
453
454     local spinopt=${1:-none}
455
```

```

456     case $spinopt in
457         aan)
458             # Maak de cursor onzichtbaar.
459             tput civis
460
461             # Oneindige loop. while :, while [ 1 -eq 1 ], of
462             # while [[ 1 -eq 1 ]] werkt ook.
463             (while true; do
464                 echo -en '\b'
465                 sleep $SPINNER_DELAY
466                 echo -en '-\b'
467                 sleep $SPINNER_DELAY
468                 echo -en '\\b'
469                 sleep $SPINNER_DELAY
470                 echo -en '| \b'
471                 sleep $SPINNER_DELAY
472             done) &
473             # Bewaar spinner-proces-identificatie.
474             SPINNER_PID=$!
475             # Zeer korte tijd tussen spinner aan en spinner uit opvangen.
476             sleep 0.1
477             ;;
478         uit)
479             [[ $SPINNER_PID ]] && kill $SPINNER_PID &> /dev/null
480             # Voorkom dat er een /, -, \, of | achterblijft.
481             echo -en ' \b'
482             # Maak de cursor weer zichtbaar.
483             tput cvvis
484             ;;
485     esac
486
487     return 0
488 }
489
490 #-Functie-----#
491 # Naam: success                                     #
492 # Doel: Tekst groen tonen naar stdout.             #
493 # Arg.: Eén optioneel argument:                   #
494 #       1. tekst      string, invoer               #
495 #-----#
496 success() {
497
498     echo -e "${S}@$${N}"
499
500     return 0
501 }
502
503 #-Functie-----#
504 # Naam: timer                                       #
505 # Doel: Bepaal de verstreken tijd ("elaps") als [DD-]UU:MM:SS.MMM. #
506 # Arg.: Twee verplichte argumenten:                #
507 #       1. starttijd  string, invoer               #
508 #                   epochseconden's'milliseconden #
509 #       1. eindtijd   string, invoer               #
510 #                   epochseconden's'milliseconden #
511 #-----#
512 timer() {
513
514     local statime=${1:-0}
515     local stotime=${2:-0}
516
517     local staepochsecs=0
518     local stoepochsecs=0
519
520     local stamillisecs=0

```



```
521     local stomillisecs=0
522
523     local -i diffcorr=0
524     local -i difftime=0
525     local -i diffdays=0
526     local -i diffhour=0
527     local -i diffmins=0
528     local -i diffsecs=0
529     local -i diffmsec=0
530
531     # Splits af de epochseconden en de milliseconden.
532     staepochsecs=$(echo $stotime | cut --delimiter='s' --fields=1)
533     stoepochsecs=$(echo $stotime | cut --delimiter='s' --fields=1)
534
535     stamillisecs=$(echo $stotime | cut --delimiter='s' --fields=2)
536     stomillisecs=$(echo $stotime | cut --delimiter='s' --fields=2)
537
538     # Verwijder voorlooppullen, bash interpreteert deze als octale waarden.
539     # vb: waarde is te groot voor basis (het onjuiste symbool is "029")
540     # De $(( )) zet een rekenkundige verband op en de 10# zet het getal om
541     # van basis 10 naar basis 10 waardoor voorlooppullen verwijderd worden.
542     stamillisecs=$(( 10#$stamillisecs ))
543     stomillisecs=$(( 10#$stomillisecs ))
544
545     # Als het aantal milliseconden bij de start groter is dan aan het einde,
546     # leen dan 1000 msec van de epochseconden en corrigeer de epochseconden.
547     if [[ $stamillisecs -gt $stomillisecs ]]; then
548         diffmsec=$(( 1000 - stamillisecs + stomillisecs ))
549         # 1000ms (=1s) geleend van epochseconden, corrigeren met 1s.
550         diffcorr=1
551     else
552         diffmsec=$(( stomillisecs - stamillisecs ))
553         # 0ms (=0s) geleend van epochseconden, corrigeren met 0s.
554         diffcorr=0
555     fi
556
557     # Verschil in seconden.
558     difftime=$(( stoepochsecs - staepochsecs - diffcorr ))
559     # Verschil in dagen.
560     diffdays=$(( difftime / 86400 ))
561     # Verschil in uren.
562     diffhour=$(( difftime % 86400 / 3600 ))
563     # Verschil in minuten.
564     diffmins=$(( difftime % 3600 / 60 ))
565     # Verschil in seconden.
566     diffsecs=$(( difftime % 60 ))
567
568     # Verwijder oude lastrunmap.
569     if [[ -d $PROGDIR/.lastrun ]]; then
570         rm $VERBOSE \
571             --recursive \
572             --force \
573             "$PROGDIR/.lastrun" |& $LOGCMD
574     fi
575
576     # Maak lastrunmap en bestand aan.
577     if [[ ! -d $LASTRUNDIR ]]; then
578         # Niveau 1 - map - éénmalig.
579         mkdir $VERBOSE \
580             --parents \
581             "$LASTRUNDIR" |& $LOGCMD
582         if $SCRIPT_NEEDS_SUDO; then
583             chown $VERBOSE \
584                 $SUDO_USER:$SUDO_USER \
585                 "$LASTRUNDIR" |& $LOGCMD
586         fi
587     fi
```

```

586         fi
587     fi
588     if [[ ! -f $LASTRUN ]]; then
589         # Niveau 2 - bestand - éénmalig per script.
590         touch "$LASTRUN" |& $LOGCMD
591         if $SCRIPT_NEEDS_SUDO; then
592             chown      $VERBOSE          \
593                     $SUDO_USER:$SUDO_USER \
594                     "$LASTRUN"          |& $LOGCMD
595         fi
596     fi
597
598     # Toon en bewaar doorlooptijd.
599     if [[ $diffdays -eq 0 ]]; then
600         printf "Script $PROGNAME duurde %02d:%02d:%02d.%03d.\r\n" \
601 $diffhour $diffmins $diffsecs $diffmsec
602         printf "Script $PROGNAME duurde de vorige keer \
603 %02d:%02d:%02d.%03d.\r\n"          \
604 $diffhour $diffmins $diffsecs $diffmsec > "$LASTRUN"
605     else
606         printf "Script $PROGNAME duurde %d-%02d:%02d:%02d.%03d.\r\n" \
607 $diffdays $diffhour $diffmins $diffsecs $diffmsec
608         printf "Script $PROGNAME duurde de vorige keer \
609 %d-%02d:%02d:%02d.%03d.\r\n"          \
610 $diffdays $diffhour $diffmins $diffsecs $diffmsec > "$LASTRUN"
611     fi
612
613     return 0
614 }
615
616 #-Functie-----#
617 # Naam: toon_gebruik #
618 # Doel: Geeft aan hoe het script aangeroepen kan worden. #
619 # Arg.: Geen argumenten. #
620 #-----#
621 toon_gebruik() {
622
623     local text=''
624
625     text="Gebruik:
626 $PROGNAME $OPTS_USAGE"
627     if $OPT_NEEDS_ARG; then
628         text="$text
629
630 Een argument dat verplicht of optioneel is voor een \
631 lange optie,
632 is dat ook voor de overeenkomstige korte optie."
633     fi
634     if $OPT_USAGE; then
635         text="$text'
636
637 Typ '$PROGNAME --help' voor meer informatie."
638     else
639         text="$text
640
641 $HELP"
642     fi
643
644     echo -e "$text" | less $LESS_OPTIONS
645
646     return 0
647 }
648
649 #-Functie-----#
650 # Naam: toon_gestart #

```

```

651 # Doel: Melding naar gebruiker en log dat script is gestart. #
652 # Arg.: Eén optioneel argument: #
653 # 1. quiet optie, invoer: #
654 # leeg verwerking met output (standaard) #
655 # gevuld verwerking zonder output (bijvoorbeeld '--quiet') #
656 #-----#
657 toon_gestart() {
658
659     local quiet=${1:-}
660
661     STATIME=$(date +%s%3N)
662
663     [[ -z $quiet ]] && info "Script $PROGNAME is gestart \
664 $(toon_tijd $STATIME).\"
665
666     log "--- Opties/argumenten (tussen > en <) ---
667 >$PARSED<
668 --- Output lsb_release ---
669 $(lsb_release --all 2> /dev/null)
670 --- Output uname ---
671 $(uname --all)
672 --- Output declare -p ---
673 $(declare -p)
674 --- Output printenv ---
675 $(printenv)\"
676
677     [[ -z $quiet ]] && [[ -f $LASTRUN ]] && info "$(cat $LASTRUN)\"
678
679     return 0
680 }
681
682 #-Functie-----#
683 # Naam: toon_gestopt #
684 # Doel: Melding naar gebruiker en log dat script is gestopt. #
685 # Arg.: Eén optioneel argument: #
686 # 1. quiet optie, invoer: #
687 # leeg verwerking met output (standaard) #
688 # --quiet verwerking zonder output #
689 #-----#
690 toon_gestopt() {
691
692     local quiet=${1:-}
693
694     local msg=''
695
696     [[ -z $quiet ]] && info
697
698     STOTIME=$(date +%s%3N)
699
700     [[ -z $quiet ]] && info "Script $PROGNAME is gestopt \
701 $(toon_tijd $STOTIME).\"
702
703     [[ -z $quiet ]] && info "$(timer $STATIME $STOTIME)\"
704
705     return 0
706 }
707
708 #-Functie-----#
709 # Naam: toon_tijd #
710 # Doel: Geef tijdstempel ("timestamp") terug in het formaat #
711 # 'op JJJJ-MM-DD om UU:MM:SS.MMM'. #
712 # Arg.: Eén verplicht argument: #
713 # 1. tijd string, invoer #
714 # (seconds since 1970-01-01 00:00:00 UTC's'milliseconds) #
715 #-----#

```

```

716 toon_tijd() {
717
718     local time=${1:-0}
719
720     local epochsecs=$(echo $time | cut --delimiter='s' --fields=1)
721     local millisecs=$(echo $time | cut --delimiter='s' --fields=2)
722     echo "$(date --date='@$epochsecs +%Y-%m-%d') \
723 $(date --date='@$epochsecs +%T').$millisecs"
724
725     return 0
726 }
727
728 #-Functie-----#
729 # Naam: toon_versie                                     #
730 # Doel: Toon versie en auteursrechten.                 #
731 # Arg.: Geen argumenten.                               #
732 #-----#
733 toon_versie() {
734
735     local copyright_years=''
736
737     info "$PROGNAME $VERSION_NUMBER"
738     if [[ $FIRST_COPYRIGHTYEAR -eq $THIS_YEAR ]]; then
739         copyright_years=$THIS_YEAR
740     else
741         copyright_years=$FIRST_COPYRIGHTYEAR-$THIS_YEAR
742     fi
743     info "Auteursrecht © $copyright_years Karel Zimmer.
744 Dit is vrije software: u mag het vrijelijk wijzigen en verder \
745 verspreiden.
746 De precieze licentie is GPL-3+: GNU General Public License versie 3 of later.
747 Zie http://gnu.org/licenses/gpl.html voor de volledige (Engelse) tekst.
748 Deze software kent GEEN GARANTIE, voor zover de wet dit toestaat.
749
750 Geschreven door Karel Zimmer, http://karelzimmer.nl, <info@karelzimmer.nl>."
751
752     return 0
753 }
754
755 #-Functie-----#
756 # Naam: verwerk_exit                                     #
757 # Doel: Verwerk de afsluitwaarde (returncode, rc) bij exit (trap), traps #
758 #       uitzetten, en debug-sessie stoppen.           #
759 # Arg.: Geen argumenten.                               #
760 #-----#
761 verwerk_exit() {
762
763     local rc=$?
764     local rc_text=''
765
766     # Negeer returncodes tussen 64 en 79; zijn geprogrammeerde fouten.
767     if [[ $rc -gt 0 && ( $rc -lt 64 || $rc -gt 79 ) ]]; then
768
769         # Stop eventuele spinner.
770         [[ $PROGNAME = 'install' || $PROGNAME = 'setup' ]] && spinner uit
771
772         # Corrigeer op ontbrekende start/stop-tijden.
773         [[ $STATIME = 0 ]] && STATIME=$(date +%s%3N)
774         [[ $STOTIME = 0 ]] && STOTIME=$(date +%s%3N)
775
776         msg="Script $PROGNAME is afgebroken $(toon_tijd $STOTIME)."
777         warning $msg
778         warning "$(timer $STATIME $STOTIME)"
779
780         # Verwerk ons bekende afsluitwaarden (returncodes).

```

```

781     case $rc in
782         $_COMMAND_NOT_EXECUTABLE)
783             rc_text=' (Opdracht gevonden maar niet uitvoerbaar)'
784             ;;
785         $_COMMAND_NOT_FOUND)
786             rc_text=' (Opdracht niet gevonden)'
787             ;;
788         $_INVALID_EXIT_ARGUMENT)
789             rc_text=' (Ongeldig exit argument)'
790             ;;
791         $_SIGPIPE)
792             rc_text=' (Zender pipe is weg, lezer pipe draait nog)'
793             ;;
794         $_INVALID_EXIT_STATUS)
795             rc_text=' (Ongeldige exit status)'
796             ;;
797         *)
798             rc_text=''
799             ;;
800     esac
801     warning "Script $PROGNAME is afgebroken met afsluitwaarde \
802 $rc$rc_text."
803     info "Controleer de eventuele meldingen hierboven.
804 Start indien gewenst script $PROGNAME opnieuw met optie debug (-d).
805 Controleer de log daarna met opdracht:"
806     local curr_exit=$(date +%Y-%m-%d %H:%M:%S')
807     command_exists systemd && \
808         info --command "journalctl -ab -t $PROGNAME -S '$curr_exit'" ||
809         info --command "grep $PROGNAME /var/log/syslog"
810
811     # Geef "fout"-geluid en (als non-root) toon schermbericht ("bubble").
812     command_exists canberra-gtk-play &&
813         canberra-gtk-play --id=suspend-error |& $LOGCMD &
814     command_exists notify-send &&
815         notify-send --urgency=critical \
816             --icon=utilities-terminal \
817             "De $PROGNAME is gestopt!" \
818             "Zie terminalvenster voor meer informatie" \
819             |& $LOGCMD
820
821 fi
822
823 # Opruimen ("clean up") voor we stoppen.
824 case $PROGNAME in
825     getstarted)
826         rm $VERBOSE \
827             --force \
828             $PROGDIR/s |& $LOGCMD
829         rm $VERBOSE \
830             --force \
831             /tmp/script-common.sh |& $LOGCMD
832     ;;
833 esac
834
835 # Sluit debug-sessie af.
836 if $OPT_DEBUG; then
837     # De xtrace stoppen.
838     set +o xtrace
839     # Sluit file descriptor 4.
840     BASH_XTRACEFD=''
841     exec 4>&-
842     # Verwijder fifo (named pipe).
843     rm $VERBOSE --force $PIPE |& $LOGCMD
844     warning "\
845     #####\n\

```

```
846 # EINDE DEBUG-SESSIE #\n\  
847 #####"  
848     info 'Controleer de log met opdracht:'  
849     local epochsecs=$(echo $STATIME | cut --delimiter='s' --fields=1)  
850     local epochexit=$(date --date="@'$epochsecs '+%Y-%m-%d %H:%M:%S')  
851     which systemd &> /dev/null &&  
852     info --command \  
853         "journalctl -ab -t $PROGNAME -S '$epochexit'" ||  
854     info --command "grep $PROGNAME /var/log/syslog"  
855 fi  
856  
857 # Trap's uitzetten.  
858 trap '' EXIT  
859 trap '' SIGINT  
860 trap '' SIGTERM  
861 trap '' SIGHUP  
862  
863 # Niet meteen exit als gestart via een starter.  
864 if $OPT_STARTER; then  
865     read -p "  
866 Druk op de Enter-toets om af te sluiten (Enter) "  
867 fi  
868  
869 # Niet meteen exit als gesourced.  
870 if [[ $PROGNAME_ORG == *bash ]]; then  
871     read -p "  
872 Druk op de Enter-toets om af te sluiten (Enter) "  
873     return 0 #XXX ????  
874 fi  
875  
876 exit $rc  
877  
878 return 0  
879 }  
880  
881 #-Functie-----#  
882 # Naam: verwerk_signal #  
883 # Doel: Handelt signaa af na trap. #  
884 # Arg.: Eén verplicht argument: #  
885 #     1. signaalspecificatie string, invoer: #  
886 #         INT     gestopt door de gebruiker met Ctrl-C (SIGINT) #  
887 #         TERM    afgebroken met de kill opdracht (SIGTERM) #  
888 #         HUP     afgebroken door het sluiten van het terminalvenster #  
889 #                 (SIGHUP) #  
890 #         *       afgebroken met een onbekend signaal #  
891 #-----#  
892 verwerk_signal() {  
893     local sig_spec=${1:-$FUNCNAME-unspecified-signal}  
894  
895     # Stop eventuele spinner.  
896     spinner uit  
897  
898     info  
899     case $sig_spec in  
900     INT)  
901         warning 'Afgebroken door de gebruiker met Ctrl-C (SIGINT).'902         ;;  
903     TERM)  
904         warning 'Afgebroken met de kill opdracht (SIGTERM).'905         ;;  
906     HUP)  
907         warning "Afgebroken door het sluiten van het terminalvenster \  
908 (SIGHUP)."  
909         ;;  
910     *)  
911         warning "Onbekend signaal ($sig_spec)."  
912         ;;  
913     esac  
914 }
```

```
911         *)
912             warning "Afgebroken met onbekend signaal ($sig_spec)."
```

---

```
913         ;;
914     esac
915
916     # Opruimen ("clean up") voor we afbreken.
917     case $PROGNAME in
918         backup)
919             if [[ $TO_DELETE ]]; then
920                 info "Opgeslagen back-up verwijderen..."
921                 rm $VERBOSE \
922                     "$TO_DELETE" |& $LOGCMD
923                 success "Opgeslagen back-up is verwijderd."
924             fi
925         ;;
926     esac
927
928     exit $E_SIGNAL_EXIT
929 }
930
931 #-Functie-----#
932 # Naam: warning #
933 # Doel: Tekst oranje/geel tonen naar stderr. #
934 # Arg.: Eén optioneel argument: #
935 #     1. tekst string, invoer #
936 #-----#
937 warning() {
938
939     echo -e "${W}@$${N}" >&2
940
941     return 0
942 }
943
944 # Einde script.
```