

```
1 #####
2 # Bestand:  script-common.sh #
3 # Doel:    Algemene variabelen en functies voor scripts. #
4 # Gebruik: . script-common.sh [NAAM] #
5 #         source script-common.sh [NAAM] #
6 # #
7 #         Opties: NAAM  gebruik NAAM als naam (PROGNAME) voor <script> #
8 #                   als deze wordt gesourced (. <script> of #
9 #                   source <script>), anders is bash de naam #
10 #                   (PROGNAME) voor <script> #
11 # Auteur:   Karel Zimmer (http://karelzimmer.nl, info@karelzimmer.nl) #
12 # ----- #
13 # Een script dient dit bestand ("library file") in te kopiëren ("include"). #
14 # Dit om te voldoen aan de zogenaamde "SPOT rule" ("Single Point of Truth #
15 # regel"): alle scripts gebruiken dezelfde variabelen en functies. #
16 # ----- #
17 # Script dient na het inlezen te vullen: #
18 # #
19 # Verplicht (aan het *begin* van Globale constanten): #
20 #     SCRIPT_NEEDS_SUDO (beheerdersrechten nodig, standaard false) #
21 #     FIRST_COPYRIGHTYEAR (eerste auteursrechtjaar, standaard dit jaar) #
22 # #
23 # Optioneel (aan het *einde* van Globale constanten): #
24 #     OPT_NEEDS_ARG      (optie met verplicht argument, standaard false) #
25 #     OPTS_SHORT        (korte opties, standaard leeg) #
26 #     OPTS_LONG         (lange opties, standaard leeg) #
27 #     OPTS_USAGE        (gebruiks-opties, standaard leeg) #
28 #     OPTS_HELP         (hulp-opties, standaard leeg) #
29 #     HELP              (hulptekst, standaard leeg, aanbevolen) #
30 # #
31 # Zie script template, en anderen, voor meer informatie. #
32 # #
33 # Enkele functienamen eindigen met '_sc' en variabele-namen met '_SC' ter #
34 # onderscheid van in een script gebruikte functie- en variabele-namen. #
35 # ----- #
36 # Dit werk valt onder een Creative Commons Naamsvermelding-GelijkDelen 4.0 #
37 # Internationaal licentie (CC BY-SA 4.0). #
38 # Bezoek http://creativecommons.org/licenses/by-sa/4.0/deed.nl om een kopie #
39 # te zien van de licentie of stuur een brief naar Creative Commons, #
40 # PO Box 1866, Mountain View, CA 94042, USA. #
41 # ----- #
42 # Versies1: 1.0.0   2009-06-17  Eerste versie. #
43 #           2.0.0   2009-11-12  Berichten in kleur. #
44 #           3.0.0   2011-01-02  Voortgangsberichten. #
45 #           4.0.0   2011-03-06  Versleutelde persoonlijke mappen (VPM). #
46 #           5.0.0   2011-11-08  Niet-versleutelde back-up/restore VPM. #
47 #           6.0.0   2012-04-22  Back-up/restore VPM als "cleartext data". #
48 #           7.0.0   2012-12-13  Vraag om [sudo] wachtwoord indien nodig. #
49 #           8.0.0   2014-10-31  Verwerk scriptargumenten zonder opties. #
50 #           9.0.0   2014-11-18  Eén logboekmap voor iedere type gebruiker. #
51 #          10.0.0   2014-12-07  Verwerk een niet-sudo gebruiker. #
52 #          11.0.0   2015-07-26  Logboek naar dagelijks i.p.v. timestamped. #
53 #          12.0.0   2015-08-05  Logboek etc. naar ~/bin i.p.v. ~/.scripts #
54 #                               en weer timestamped. #
55 #          13.0.0   2015-10-10  Functie verwerk_rc van 2 naar 3 parameters. #
56 #          14.0.0   2016-05-20  Van eigen log naar system log en systemd #
57 #                               journal. #
58 #          15.0.0   2017-03-20  Ontleden opties en argumenten door getopt. #
59 #          16.0.0   2017-06-01  Met optie debug opdrachtinformatie en #
60 #                               scriptverloop naar de log. #
61 # #
62 # ----- #
63 # 1 Toelichting versienummering en -datum: #
64 #     V.R.M  VER.-DATUM  Versie.Release.Modificatienummer en datum #
65 #     =.=.=  =====  Alleen commentaar of spatiëring gewijzigd. #
```

```

66 #   =.+   JJJJ-MM-DD   Kleine technische wijziging.           #
67 #   =.+.0 JJJJ-MM-DD   Kleine functionele wijziging.        #
68 #   +.0.0 JJJJ-MM-DD   Nieuwe functionaliteit.              #
69 #####
70 # VERSION_NUMBER=16.1.1           # Versienummer
71 # RELEASE_DATE=2017-06-20         # Versiedatum
72
73 #####
74 # Instellingen                                     #
75 #####
76 set -o pipefail                                   # Pipe eindigt met hoogste returncode
77
78 #-----#
79 # Algemene globale constanten                       #
80 #-----#
81
82 # Algemeen verplicht - standaardwaarden =====
83 # (aan het *begin* van Globale constanten) =====
84 declare SCRIPT_NEEDS_SUDO=false                 # Standaard geen beheerdersrechten
85                                                    # nodig
86 declare FIRST_COPYRIGHTYEAR=2017               # Standaard eerste auteursrechtjaar
87
88 # Algemeen =====
89 readonly ASK_FOR_PW_HELPTEXT="Indien nodig wordt het beheerderswachtwoord \
90 gevraagd."                                     # Vraag om beheerderswachtwoord
91 readonly BINDIR=$HOME/bin                       # Direct-uitvoerbare-scriptsmap
92 readonly DASHES=$(printf '\u002D%.0s' {1..77}) # Streepregel, Unicode002D='- '
93                                                    # Ontwikkelmachine
94 readonly DEVMACH='PC06'                         # Ontwikkelaar
95 readonly DEVUSER='karel'
96 declare DISTID=$(lsb_release --id --short)      # Standaard distributie-ID,
97                                                    # bijvoorbeeld Ubuntu
98
99 readonly DISTREL=$(lsb_release --release --short) # Distributieverisie, bijv. 12.04
100                                                    # Distributie-indicatoren:
101
102 declare DEBIAN_CINNAMON=false
103 declare DEBIAN=false                            # Standaard GNOME
104 declare DEBIAN_KDE=false
105 declare DEBIAN_LXDE=false
106 declare DEBIAN_MATE=false
107 declare DEBIAN_XFCE=false
108 declare KUBUNTU=false
109 declare LUBUNTU=false
110 declare UBUNTU=false                            # Unity t/m 17.04, GNOME v/a 17.10
111 declare UBUNTU_GNOME=false                      # T/m 17.04
112 declare UBUNTU_MATE=false
113 declare XUBUNTU=false
114 case $XDG_CURRENT_DESKTOP in
115     X-Cinnamon)
116         if [[ $DISTID = 'Debian' ]]; then
117             DISTID='Debian-Cinnamon'
118             readonly DEBIAN_CINNAMON=true
119         fi
120     ;;
121     GNOME)
122         if [[ $DISTID = 'Debian' ]]; then
123             DISTID='Debian'
124             readonly DEBIAN=true
125         fi
126         if [[ $DISTID = 'Ubuntu' ]]; then
127             if [[ $DISTREL < 17.10 ]]; then
128                 DISTID='Ubuntu-GNOME'
129                 readonly UBUNTU_GNOME=true
130             else

```

```

131         DISTID='Ubuntu'
132         readonly UBUNTU=true
133     fi
134 fi
135 ;;
136 KDE)
137     if [[ $DISTID = 'Debian' ]]; then
138         DISTID='Debian-KDE'
139         readonly DEBIAN_KDE=true
140     fi
141     if [[ $DISTID = 'Ubuntu' ]]; then
142         DISTID='Ubuntu-KDE'
143         readonly KUBUNTU=true
144     fi
145 ;;
146 LXDE)
147     if [[ $DISTID = 'Ubuntu' ]]; then
148         DISTID='Lubuntu'
149         readonly LUBUNTU=true
150     fi
151 ;;
152 MATE)
153     if [[ $DISTID = 'Debian' ]]; then
154         DISTID='Debian-Mate'
155         readonly DEBIAN_MATE=true
156     fi
157     if [[ $DISTID = 'Ubuntu' ]]; then
158         DISTID='Ubuntu-MATE'
159         readonly UBUNTU_MATE=true
160     fi
161 ;;
162 Unity)
163     DISTID='Ubuntu'
164     readonly UBUNTU=true
165 ;;
166 XFCE)
167     if [[ $DISTID = 'Ubuntu' ]]; then
168         DISTID='Xubuntu'
169         readonly XUBUNTU=true
170     fi
171 ;;
172 *)
173     : # Onbekende DE, DISTID blijft gelijk aan de standaard.
174 ;;
175 esac
176 readonly DISTID                # Distributie-ID, bijvoorbeeld Ubuntu
177 command -v startx &> /dev/null &&
178     readonly EDITION='desktop' ||
179     readonly EDITION='server'   # Editie
180     readonly OPTS_SHORT_SC='huvd' # Korte opties
181     readonly OPTS_LONG_SC='help,usage,version,debug'
182                                     # Lange opties
183     readonly OPTS_USAGE_SC=$(cat << OPTS_USAGE_SC
184 [-h|--help] [-u|--usage] [-v|--version] [-d|--debug]
185 OPTS_USAGE_SC
186 )                                     # Gebruiks-opties
187     readonly OPTS_HELP_SC=$(cat << OPTS_HELP_SC
188 -h --help           Deze hulptekst tonen en stoppen
189 -u --usage          Een korte gebruikssamenvatting tonen en stoppen
190 -v --version        Scriptversie tonen en stoppen
191 -d --debug          Opdrachtinformatie en scriptverloop naar de log
192 OPTS_HELP_SC
193 )                                     # Help-opties
194 readonly PART_OF_INSTALL_HELPTEXT=$(cat << PART_OF_INSTALL_HELPTEXT
195 Dit script wordt ook gebruikt bij het installeren van Linux.

```

```

196 Ga voor meer informatie naar de site http://karelzimmer.nl en
197 kijk naar Checklist installatie.
198 PART_OF_INSTALL_HELPTEXT
199 ) # Installatietekst
200 # Bij ssh-connectie kan $0 '-bash' opleveren, vandaar na --.
201 readonly PROG=$(readlink --canonicalize -- "$0")
202 # Absoluut script (bijvoorbeeld
203 # geen './gset' maar '/tmp/gset')
204 readonly PROGDIR_ORG=$(dirname "$PROG") # Scriptmap oorspronkelijk
205 readonly PROGNAME_ORG=$(basename "$PROG")
206 # Scriptnaam oorspronkelijk
207 declare PROGDIR=$PROGDIR_ORG # Scriptmap standaard
208 [[ $PROGNAME_ORG = 'bash' || $PROGNAME_ORG = '-bash' ]] && PROGDIR=$(pwd)
209 # Scriptmap als gesourced
210 readonly PROGDIR # Scriptmap
211 declare PROGNAME=$PROGNAME_ORG # Scriptnaam standaard
212 [[ $PROGNAME_ORG = 'bash' || $PROGNAME_ORG = '-bash' ]] &&
213 PROGNAME=${1:-$PROGNAME_ORG} # Scriptnaam als gesourced
214 readonly PROGNAME # Scriptnaam
215
216 readonly CMDFILE_HELPTEXT=$(cat << CMDFILE_HELPTEXT
217 De te verwerken bestanden:
218 - dienen te beginnen met '$PROGNAME-',
219 - bevatten commentaar- en opdrachtregels, waarbij:
220 1. een commentaarregel die begint met:
221 # of spatie - wordt overgeslagen,
222 #1 - wordt getoond als voortgang,
223 #2 - wordt getoond als informatie voor \
224 installeren/instellen van de software,
225 #3 - wordt getoond als informatie voor verwijderen van de \
226 software.
227 2. de opdrachtregels worden verwerkt op de volgorde zoals deze in het
228 bestand voorkomen.
229 CMDFILE_HELPTEXT
230 ) # Hulptekst opdrachtbestand
231 readonly HELPLINE="Typ '$PROGNAME --usage' of '$PROGNAME --help' voor meer \
232 informatie." # Hulpregel
233 readonly LESS_OPTIONS='-FRSXMK' # Less-opties, overgenomen van
234 # systemctl en journalctl, zie bijv.
235 # 'man systemctl', zoek $SYSTEMD_LESS
236 command -v systemd &> /dev/null &&
237 readonly LOGCMD="systemd-cat --identificer=$PROGNAME --priority=info" ||
238 readonly LOGCMD="logger --tag $PROGNAME --priority info"
239 # Logging-opdracht⊗
240 # ⊗ Toelichting system log en systemd journal:
241 # - Logging naar systemd journal op systemd-systemen met:
242 # <opdracht> |& systemd-cat -t $PROGNAME -p info
243 # Info uit de journal met:
244 # journalctl -ab -t $PROGNAME [-S <tijd>]
245 # - Logging naar system log op non-systemd-systemen met:
246 # <opdracht> |& logger -t $PROGNAME -p info
247 # Info uit de log met:
248 # grep $PROGNAME /var/log/syslog
249 readonly OUTDIR1=$HOME/.output # Uitvoermap niveau 1 (éénmalig)
250 readonly OUTDIR2=$OUTDIR1/$PROGNAME # Uitvoermap niveau 2 (1malig/script)
251 readonly SPACES=$(printf '%.0s' {1..77})
252 # Spatieregel, ' '
253 readonly SUDO_PROMPT_SC=$(cat << SUDO_PROMPT_SC
254 Script $PROGNAME moet worden uitgevoerd met beheerdersrechten.
255 sudo $PROGDIR/$PROGNAME $@
256 [sudo] wachtwoord voor %p:
257 SUDO_PROMPT_SC
258 ) # Prompt als beheerder
259 readonly TIMESTAMP=$(date +%Y-%m-%d-%H.%M.%S.%N)
260 # Tijdstempel JJJJ-MM-DD-UU.MM.SS.MMM

```

```

261                                     # Lees gebruikersmapnamen in:
262 if ! source ${XDG_CONFIG_HOME:-$HOME/.config}/user-dirs.dirs &> /dev/null
263 then
264     if locale | grep --quiet '=nl'; then
265         readonly XDG_DESKTOP_DIR=$HOME/Bureaublad
266         readonly XDG_DOWNLOAD_DIR=$HOME/Downloads
267         readonly XDG_TEMPLATES_DIR=$HOME/Sjablonen
268         readonly XDG_PUBLICSHARE_DIR=$HOME/Openbaar
269         readonly XDG_DOCUMENTS_DIR=$HOME/Documenten
270         readonly XDG_MUSIC_DIR=$HOME/Muziek
271         readonly XDG_PICTURES_DIR=$HOME/Afbeeldingen
272         readonly XDG_VIDEOS_DIR="$HOME/Video's"
273     else
274         readonly XDG_DESKTOP_DIR=$HOME/Desktop
275         readonly XDG_DOWNLOAD_DIR=$HOME/Downloads
276         readonly XDG_TEMPLATES_DIR=$HOME/Templates
277         readonly XDG_PUBLICSHARE_DIR=$HOME/Public
278         readonly XDG_DOCUMENTS_DIR=$HOME/Documents
279         readonly XDG_MUSIC_DIR=$HOME/Music
280         readonly XDG_PICTURES_DIR=$HOME/Pictures
281         readonly XDG_VIDEOS_DIR=$HOME/Videos
282     fi
283 fi
284
285 # Tekstattributen, zie ook: man terminfo. Tekst <opmaak>      <functie>
286 readonly N=$(tput sgr0)           # Tekst normaal      info
287 readonly B=$(tput bold)           # Tekst vet          info --bold
288 readonly C=${B}${tput setaf 4}    # Tekst blauw       info --command
289 readonly I=${tput sitm}           # Tekst schuin      info --italic
290 readonly S=${B}${tput setaf 2}    # Tekst groen       success
291 tput longname | grep --quiet 256  &&
292 readonly W=${B}${tput setaf 208}  || # Tekst oranje     warning
293 readonly W=${B}${tput setaf 3}    # Tekst geel        warning
294 readonly E=${B}${tput setaf 1}    # Tekst rood        error
295 # Voorbeeld kleurgebruik:
296 #   Altijd <opmaak> zodat het duidelijk is dat het opmaak betreft.
297 #   - echo -e "${E}Foutje${N}"
298 #   - echo -en {E}; echo 'Foutje'; echo -en ${N}
299 #   (voordeel is dat dan er geen stuurtekens zichtbaar zijn)
300 #   - voorbeeld gebruik functie: error 'Foutje'
301
302                                     # Diverse statussen:
303 readonly STATUS_SIMULATE="[${W}SIMULATIE${N}]"
304 readonly STATUS_SUCCESS="[${S}GOED${N}]"
305 readonly STATUS_ERROR="[${E}FOUT${N}]"
306 readonly TEXT_SIMULATE="[${W}SIMULATIE${N}]"
307
308 readonly BACKUP_CONTENT_HELPTEXT=$(cat << BACKUP_CONTENT_HELPTEXT
309 De inhoud van de back-up is op te vragen met opdracht:
310 ${C}tar tvf ${I}back-upbestand${N}${C} [--wildcards SELECTIE...])${N}
311 BACKUP_CONTENT_HELPTEXT
312 )                                     # Inhoud back-upbestand hulptekst
313
314 # Specifiek =====
315 readonly GPL=$(cat << GPL
316 Dit is vrije software: u mag het vrijelijk wijzigen en verder verspreiden.
317 De precieze licentie is GPL-3+: GNU General Public License versie 3 of later.
318 Zie http://gnu.org/licenses/gpl.html voor de volledige (Engelse) tekst.
319 Deze software kent GEEN GARANTIE, voor zover de wet dit toestaat.
320 GPL
321 )                                     # GPL licentietekst
322 readonly LASTRUNDIR=$HOME/.lastrun   # Laatste-uitvoeringsmap
323 readonly LASTRUN=$LASTRUNDIR/lastrun-$PROGNAME
324                                     # Laatste-uitvoeringsbestand
325 readonly SCNAME='script-common'     # Script-common-naam

```

```
326 readonly SUPERUID=0 # UserID beheerder, superuser, root
327 readonly THIS_YEAR=$(date +%Y) # Dit jaar (JJJJ)
328 readonly USERS=$(
329     while read record; do
330         grp=$(echo "$record" | awk -F ':' '{print $4}')
331         if [[ $grp -ge 1000 && $grp -ne 65534 ]]; then
332             echo "$record" | awk -F ':' '{print $1}'
333         fi
334     done < /etc/passwd
335 ) # Gebruikers
336 [[ -d /home/.ecryptfs ]] && VPM=true || VPM=false
337 readonly VPM # VPM in gebruik?
338 [[ $(echo $USERS | wc --words) -gt 1 ]] && USR2=true || USR2=false
339 readonly USR2 # Meer dan één gebruiker?
340 readonly W_ECRYPTFS_DETECTED=$(cat << W_ECRYPTFS_DETECTED
341 #####
342
343 WAARSCHUWING:
344
345 Er zijn versleutelde persoonlijke mappen in gebruik en meer dan één gebruiker
346 op deze computer.
347
348 Voer de $PROGNAME uit per gebruiker !
349
350 Gebruikers van deze computer zijn:
351 $USERS
352
353 De $PROGNAME wordt nu uitgevoerd voor:
354 ${SUDO_USER:-$USER}
355
356 #####
357 W_ECRYPTFS_DETECTED
358 ) # eCryptfs waarschuwing
359
360 # Algemeen optioneel - standaardwaarden =====
361 # (aan het *einde* van Globale constanten) =====
362 # Voor script die alleen script-common.sh en controleer_invoer_sc gebruikt. =
363 declare OPT_NEEDS_ARG=false # Standaard geen optie met een
364 # verplicht argument
365 declare OPTS_SHORT='' # Standaard korte opties
366 declare OPTS_LONG='' # Standaard lange opties
367 declare OPTS_HELP=$(cat << OPTS_HELP
368 OPTS_HELP
369 ) # Standaard hulp-opties
370 declare OPTS_USAGE=$(cat << OPTS_USAGE
371 OPTS_USAGE
372 ) # Standaard gebruiks-opties
373 declare HELP=$(cat << HELP
374
375 $OPTS_HELP_SC$OPTS_HELP
376
377 Script $PROGNAME verstrekt verder geen hulp-informatie.
378 HELP
379 ) # Standaard hulptekst
380
381 # Foutcodes =====
382 readonly E_SUDO_NOT_ALLOWED=100
383 readonly E_WE_ARE_SUPERUSER=101
384 readonly E_TASK_ERROR=102
385 readonly E_SIGNAL_EXIT=103
386 readonly E_COMMAND_NOT_EXECUTABLE=126
387 readonly E_COMMAND_NOT_FOUND=127
388 readonly E_INVALID_EXIT_ARGUMENT=128
389 readonly E_INVALID_EXIT_STATUS=255
390
```

```

391 #-----#
392 # Algemene globale variabelen #
393 #-----#
394
395 # Array, integer =====
396
397 # Boolean =====
398 declare OPT_DEBUG=false # Optie debug opgegeven
399 declare OPT_HELP=false # Optie help opgegeven
400 declare OPT_STARTER=false # Optie starter (in desktopbestand)
401 declare OPT_USAGE=false # Optie usage opgegeven
402 declare OPT_VERSION=false # Optie version opgegeven
403 declare SCRIPT_KILLED=false # Script is afgebroken
404
405 # Overig =====
406 declare PARSED='' # Door getopt ontlede opties/argum.
407 declare SPINNER_DELAY=0.1 # Tijd tussen spinafbeeldingen
408 declare SPINNER_PID='' # Spinner-proces-ID
409 declare STATIME=0 # Timer's starttijd
410 declare STOTIME=0 # Timer's eindtijd
411 declare TO_DELETE='' # Evt. te verwijderen back-up
412 declare VERBOSE='' # Meer opdrachtinformatie
413
414 #####
415 # Algemene functies (op alfabetische volgorde) #
416 #####
417
418 #-Functie-----#
419 # Naam: attendeer_gebruiker #
420 # Doel: Geef "attentie"-geluid en optioneel vraag gebruiker om reactie. #
421 # Arg.: Eén optioneel argument: #
422 # 1. tekst string, invoer #
423 #-----#
424 attendeer_gebruiker() {
425     log "$SCNAME:$FUNCNAME:$LINENO:$@"
426
427     canberra-gtk-play --id=message \
428                     --loop=3 \
429                     --volume=-20 |& $LOGCMD &
430
431     if [[ $@ ]]; then
432         info
433         read -p "$@"
434     fi
435
436     return 0
437 }
438
439 #-Functie-----#
440 # Naam: command_exists #
441 # Doel: Controleer of opdracht beschikbaar is op het systeem. #
442 # Arg.: Geen argumenten #
443 # Afsluitwaarde: 0 opdracht is beschikbaar #
444 # 1 opdracht is niet beschikbaar #
445 #-----#
446 command_exists() {
447     log "$SCNAME:$FUNCNAME:$LINENO:$@"
448     command -v "$@" &> /dev/null
449 }
450
451 #-Functie-----#
452 # Naam: controleer_gebruiker_sc #
453 # Doel: Bekijkt of de gebruiker de juiste rechten heeft. #
454 # Arg.: Twee verplichte argumenten: #
455 # 1. script_needs_sudo boolean, invoer: #

```

```
456 #         true   we moeten beheerder (superuser of root) zijn           #
457 #         false  we moeten standaardgebruiker zijn                       #
458 #         2. Script-opties en -argumenten.                               #
459 #-----#
460 controleer_gebruiker_sc() {
461     log "$SCNAME:$FUNCNAME:$LINENO:$@"
462
463     local script_needs_sudo=${1:-false}
464     local grp=''
465
466     # We moeten root zijn, controleer of dat zo is.
467     if $script_needs_sudo && [[ $UID -ne $SUPERUID ]]; then
468
469         # Maak een lijst van toegestane (en verboden) opdrachten voor de
470         # gebruiker. Als dit goed gaat mag de gebruiker sudo gebruiken en de
471         # lijst niet tonen. Als dit fout gaat mag de gebruiker geen sudo
472         # gebruiken, toon foutmelding, geef meer informatie en toon
473         # gebruikers.
474         sudo --list \
475             --prompt \
476             "$SUDO_PROMPT_SC" | $LOGCMD
477         if [[ $? -gt 0 ]]; then
478             info "Om script $PROGNAME te kunnen uitvoeren zijn \
479 beheerdersrechten nodig,"
480             info "met 'sudo', of als root."
481             info 'Mogelijke gebruikers met beheerdersrechten zijn:'
482             info "$USERS"
483             info
484             if $OPT_STARTER; then
485                 if [[ $PROGNAME = 'backup' ]]; then
486                     info "Gebruik starter 'Backup --no-root uitvoeren' om \
487 alleen de eigen Persoonlijke map veilig te stellen."
488                 elif [[ $PROGNAME = 'restore' ]]; then
489                     info "Gebruik starter 'Restore --no-root uitvoeren' om \
490 alleen de eigen Persoonlijke map terug te zetten."
491                 fi
492             else
493                 info "Typ bijvoorbeeld 'su beheerder', waarbij beheerder \
494 voldoende rechten heeft,"
495                 info "of meld u aan als beheerder, en voer daarna script \
496 $PROGNAME opnieuw uit."
497             fi
498             exit_sc $E_SUDO_NOT_ALLOWED
499         fi
500
501         # Start het script opnieuw, maar dan met 'sudo'.
502         shift # Eerste argument is verwerkt
503         log "Start script $PROGDIR/$PROGNAME opnieuw met sudo..."
504         sudo --preserve-env $PROGDIR/$PROGNAME "$@"
505         exit $?
506     fi
507
508     # We moeten niet-root zijn, controleer of dat zo is.
509     if ! $script_needs_sudo && [[ $UID -eq $SUPERUID ]]; then
510
511         [[ $PROGNAME = 'getstarted' ]] &&
512             rm $VERBOSE $PROGDIR/s |& $LOGCMD
513         [[ $PROGNAME = 'getstarted' ]] &&
514             rm $VERBOSE /tmp/script-common.sh |& $LOGCMD
515
516         error "Script $PROGNAME mag niet met beheerdersrechten (met 'sudo', \
517 of als root) worden uitgevoerd."
518         exit_sc $E_WE_ARE_SUPERUSER
519     fi
520
```



```

521     shift    # Eerste argument is verwerkt
522
523     # Trap pas hier, anders dubbele "trap signal" bij een root-script die
524     # gestart wordt als een standaardgebruiker i.p.v. beheerder. Trap niet
525     # uitvoeren als aanroepend <script> is gesourced (. <script> of
526     # source <script>).
527     [[ $PROGNAME_ORG = 'bash' || $PROGNAME_ORG = '-bash' ]] && return 0
528
529     trap "verwerk_signal_exit INT" INT
530     trap "verwerk_signal_exit TERM" TERM
531     trap "verwerk_signal_exit HUP" HUP
532
533     return 0
534 }
535
536 #-Functie-----#
537 # Naam: controleer_invoer_sc #
538 # Doel: Controleer, en soms verwerk, de opties en argumenten van het script.#
539 # Arg.: Script-opties en -argumenten. #
540 #-----#
541 controleer_invoer_sc() {
542     log "$SCNAME:$FUNCNAME:$LINENO:$@"
543
544     local comma=''
545     local copyright_years=''
546     local -i exclude_arg=0
547     local -i file_arg=0
548     local -i search4_arg=0
549     local -i select_arg=0
550     local parm_error=false
551
552     # Test op aanwezigheid nieuwe versie getopt.
553     getopt --test > /dev/null
554     if [[ $? -eq 4 ]]; then
555         # Vul komma of niets voor getopt's longoptions.
556         [[ $OPTS_LONG ]] && comma=',' || comma=''
557
558         # Ontleed (parse) opties en argumenten met getopt.
559         PARSED=$(getopt --alternative \
560             --options $OPTS_SHORT_SC$OPTS_SHORT \
561             --longoptions $OPTS_LONG_SC$comma$OPTS_LONG \
562             --name "$PROGNAME" \
563             -- "$@")
564
565         # Controleer op fouten.
566         if [[ $? -gt 0 ]]; then
567             PARSED=''
568             parm_error=true
569         fi
570
571         # Gebruik eval om quoting correct te verwerken.
572         eval set -- "$PARSED"
573     else
574         log "$PROGNAME: oude versie van getopt in gebruik; opties opgeven \
575 volgens optie usage (-u)"
576     fi
577
578     # Evalueer de opties en argumenten meegegeven bij de aanroep.
579     while [[ $# ]]; do
580
581         # Verwerk eerst de algemene opties en argumenten. De standaard
582         # opties (true/false) staan in dit script.
583         case $1 in
584             -h|--help)
585                 OPT_HELP=true

```

```
586         toon_gebruik_sc $OPT_USAGE
587         exit_sc 0
588         ;;
589     -u|--usage)
590         OPT_USAGE=true
591         toon_gebruik_sc $OPT_USAGE
592         exit_sc 0
593         ;;
594     -d|--debug)
595         OPT_DEBUG=true
596         VERBOSE='--verbose'
597         shift
598         continue
599         ;;
600     -v|--version)
601         OPT_VERSION=true
602         toon_versie
603         exit_sc 0
604         ;;
605     --)
606         # Einde van de opties, nu alleen nog maar argumenten.
607         shift
608         continue
609         ;;
610 esac
611
612 # Verwerk nu de script-opties en -argumenten. De standaard opties/
613 # argumenten (true/false) staan in het betreffende script.
614 case $PROGNAME in
615     archive)
616         case $1 in
617             --delete)
618                 OPT_DELETE=true
619                 shift
620                 continue
621                 ;;
622             *) # Argument archiefbestand.
623                 ARG_SEARCH4=true
624                 SEARCH4_ARG[$search4_arg]=$1
625                 (( search4_arg = search4_arg + 1 ))
626                 shift
627                 continue
628                 ;;
629         esac
630         ;;
631     backup)
632         case $1 in
633             -n|--no-root)
634                 OPT_NO_ROOT=true
635                 SCRIPT_NEEDS_SUDO=false
636                 shift
637                 continue
638                 ;;
639             -t|--target)
640                 if $OPT_TARGET; then
641                     info "$PROGNAME: teveel opties target (-t)"
642                     parm_error=true
643                 else
644                     OPT_TARGET=true
645                     TARGET_ARG=$2
646                 fi
647                 shift 2
648                 continue
649                 ;;
650             --starter)
```

```
651         OPT_STARTER=true
652         shift
653         continue
654         ;;
655     -x|--exclude)
656         OPT_EXCLUDE=true
657         EXCLUDE_ARG[$exclude_arg]=$2
658         (( exclude_arg = exclude_arg + 1 ))
659         shift 2
660         continue
661         ;;
662     *) # Argument selectie.
663         ARG_SELECT=true
664         SELECT_ARG[$select_arg]=$1
665         (( select_arg = select_arg + 1 ))
666         shift
667         continue
668         ;;
669     esac
670     ;;
671     ckname)
672     case $1 in
673     *) # Argument startlocatie.
674         if $ARG_START; then
675             info "$PROGNAME: teveel argumenten"
676             parm_error=true
677         else
678             ARG_START=true
679             START_ARG=$1
680         fi
681         shift
682         continue
683         ;;
684     esac
685     ;;
686     ckinst)
687     case $1 in
688     -l|--list)
689         OPT_LIST=true
690         shift
691         continue
692         ;;
693     *) # Argument installatie/instellingsbestand.
694         ARG_FILE=true
695         FILE_ARG[$file_arg]=$1
696         (( file_arg = file_arg + 1 ))
697         shift
698         continue
699         ;;
700     esac
701     ;;
702     deploy)
703     case $1 in
704     -c|--checklists)
705         OPT_CHECKLISTS=true
706         shift
707         continue
708         ;;
709     -e|--edit)
710         OPT_EDIT=true
711         shift
712         continue
713         ;;
714     -f|--force)
715         OPT_FORCE=true
```

```
716         shift
717         continue
718         ;;
719     -l|--list)
720         OPT_LIST=true
721         shift
722         continue
723         ;;
724     -p|--print)
725         OPT_PRINT=true
726         shift
727         continue
728         ;;
729     -r|--reset)
730         OPT_RESET=true
731         shift
732         continue
733         ;;
734     -s|--scripts)
735         OPT_SCRIPTS=true
736         shift
737         continue
738         ;;
739     esac
740     ;;
741     freerdp)
742         case $1 in
743             --starter)
744                 OPT_STARTER=true
745                 shift
746                 continue
747                 ;;
748             esac
749             ;;
750     install)
751         case $1 in
752             -i|--install-info)
753                 OPT_INSTALL_INFO=true
754                 shift
755                 continue
756                 ;;
757             -j|--uninstall-info)
758                 OPT_UNINSTALL_INFO=true
759                 shift
760                 continue
761                 ;;
762             -l|--list)
763                 OPT_LIST=true
764                 shift
765                 continue
766                 ;;
767             -s|--simulate)
768                 OPT_SIMULATE=true
769                 shift
770                 continue
771                 ;;
772             --starter)
773                 OPT_STARTER=true
774                 shift
775                 continue
776                 ;;
777             *) # Argument installatiebestand.
778                 ARG_FILE=true
779                 FILE_ARG[$file_arg]=$1
780                 (( file_arg = file_arg + 1 ))
```

```
781             shift
782             continue
783             ;;
784         esac
785         ;;
786     lshost)
787         case $1 in
788             -a|--all)
789                 OPT_ALL=true
790                 shift
791                 continue
792                 ;;
793             -e|--edit)
794                 OPT_EDIT=true
795                 shift
796                 continue
797                 ;;
798         esac
799         ;;
800     list-installed)
801         case $1 in
802             --starter)
803                 OPT_STARTER=true
804                 shift
805                 continue
806                 ;;
807         esac
808         ;;
809     mkusb)
810         case $1 in
811             -s|--source)
812                 if $OPT_SOURCE; then
813                     info "$PROGNAME: teveel opties source (-s)"
814                     parm_error=true
815                 else
816                     OPT_SOURCE=true
817                     SOURCE_ARG=$2
818                 fi
819                 shift 2
820                 continue
821                 ;;
822             *) # Argument USB-stick-device.
823                 if $ARG_DEVICE; then
824                     info "$PROGNAME: teveel argumenten"
825                     parm_error=true
826                 else
827                     ARG_DEVICE=true
828                     DEVICE_ARG=$1
829                 fi
830                 shift
831                 continue
832                 ;;
833         esac
834         ;;
835     nas)
836         case $1 in
837             --starter)
838                 OPT_STARTER=true
839                 shift
840                 continue
841                 ;;
842         esac
843         ;;
844     printer-install)
845         case $1 in
```

```
846         -i|--printer-info)
847             OPT_PRINTER_INFO=true
848             shift
849             continue
850             ;;
851         -s|--simulate)
852             OPT_SIMULATE=true
853             shift
854             continue
855             ;;
856         --starter)
857             OPT_STARTER=true
858             shift
859             continue
860             ;;
861     esac
862     ;;
863 restore)
864     case $1 in
865         -n|--no-root)
866             OPT_NO_ROOT=true
867             SCRIPT_NEEDS_SUDO=false
868             shift
869             continue
870             ;;
871         -s|--source)
872             if $OPT_SOURCE; then
873                 info "$PROGNAME: teveel opties source (-s)"
874                 parm_error=true
875             else
876                 OPT_SOURCE=true
877                 SOURCE_ARG=$2
878             fi
879             shift 2
880             continue
881             ;;
882         -t|--target)
883             if $OPT_TARGET; then
884                 info "$PROGNAME: teveel opties target (-t)"
885                 parm_error=true
886             else
887                 OPT_TARGET=true
888                 TARGET_ARG=$2
889             fi
890             shift 2
891             continue
892             ;;
893         -x|--exclude)
894             OPT_EXCLUDE=true
895             EXCLUDE_ARG[$exclude_arg]=$2
896             (( exclude_arg = exclude_arg + 1 ))
897             shift 2
898             continue
899             ;;
900         --starter)
901             OPT_STARTER=true
902             shift
903             continue
904             ;;
905         *) # Argument selectie.
906             ARG_SELECT=true
907             SELECT_ARG[$select_arg]=$1
908             (( select_arg = select_arg + 1 ))
909             shift
910             continue
```

```
911         ;;
912     esac
913     ;;
914 rninst)
915     case $1 in
916     -l|--list)
917         OPT_LIST=true
918         shift
919         continue
920         ;;
921     *) # Argument installatie/instellingsbestand.
922         ARG_FILE=true
923         FILE_ARG[$file_arg]=$1
924         (( file_arg = file_arg + 1 ))
925         shift
926         continue
927         ;;
928     esac
929     ;;
930 setup)
931     case $1 in
932     -i|--setup-info)
933         OPT_SETUP_INFO=true
934         shift
935         continue
936         ;;
937     -j|--unsetup-info)
938         OPT_UNSETUP_INFO=true
939         shift
940         continue
941         ;;
942     -l|--list)
943         OPT_LIST=true
944         shift
945         continue
946         ;;
947     -s|--simulate)
948         OPT_SIMULATE=true
949         shift
950         continue
951         ;;
952     --starter)
953         OPT_STARTER=true
954         shift
955         continue
956         ;;
957     *) # Argument instellingsbestand.
958         ARG_FILE=true
959         FILE_ARG[$file_arg]=$1
960         (( file_arg = file_arg + 1 ))
961         shift
962         continue
963         ;;
964     esac
965     ;;
966 start-install)
967     case $1 in
968     -s|--simulate)
969         OPT_SIMULATE=true
970         shift
971         continue
972         ;;
973     esac
974     ;;
975 template)
```

```
976             break
977             ;;
978         wallpapers)
979             case $1 in
980                 --starter)
981                     OPT_STARTER=true
982                     shift
983                     continue
984                 ;;
985             esac
986             ;;
987         esac
988     done
989
990     # Geef optie gebruik en help aan als er fouten zijn geconstateerd.
991     if $parm_error; then
992         info $HELPLINE
993         exit_sc 1
994     fi
995
996     return 0
997 }
998
999 #-Functie-----#
1000 # Naam: error #
1001 # Doel: Tekst rood tonen en naar de log/journal sturen. #
1002 # Arg.: Eén optioneel argument: #
1003 #     1. tekst      string, invoer #
1004 #-----#
1005 error() {
1006     echo -e "${E}${@}$N}" >&2
1007     $OPT_DEBUG && log --error "$@"
1008     return 0
1009 }
1010
1011 #-Functie-----#
1012 # Naam: exit_sc #
1013 # Doel: Niet meteen exit als gestart via een starter, en return in plaats #
1014 #       van exit als gesourced. #
1015 # Arg.: Eén optioneel argument: #
1016 #     1. afsluitwaarde integer, invoer #
1017 #-----#
1018 exit_sc() {
1019     local rc=${1:-0}
1020
1021     if $OPT_STARTER; then
1022         info
1023         read -p 'Druk op de Enter-toets om af te sluiten (Enter) '
1024     fi
1025
1026     [[ $PROGNAME = 'getstarted' ]] && rm $VERBOSE $PROGDIR/s |& $LOGCMD
1027     [[ $PROGNAME_ORG = 'bash' || $PROGNAME_ORG = '-bash' ]] &&
1028     return $rc ||
1029     exit $rc
1030 }
1031
1032 #-Functie-----#
1033 # Naam: info #
1034 # Doel: Tekst tonen, eventueel vet (extra opvallend) of blauw (voor #
1035 #       opdrachten), en naar log/journal sturen. #
1036 # Arg.: Twee optionele argumenten: #
1037 #     1. optie      string, invoer: #
```



```
1041 #           -b, --bold, -i, --italic -c, --command           #
1042 #           2. tekst      string, invoer                       #
1043 #-----#
1044 info() {
1045     local optie=${N}
1046     local argum=''
1047
1048     while [[ @$@ ]]; do
1049         case $1 in
1050             -b|--bold)
1051                 optie=${B}
1052                 shift
1053                 continue
1054                 ;;
1055             -c|--command)
1056                 optie=${C}
1057                 shift
1058                 continue
1059                 ;;
1060             -i|--italic)
1061                 optie=${I}
1062                 shift
1063                 continue
1064                 ;;
1065             *) # Argument tekst.
1066                 if [[ -z $argum ]]; then
1067                     argum=$1
1068                 else
1069                     argum="$argum $1"
1070                 fi
1071                 shift
1072                 continue
1073                 ;;
1074         esac
1075     done
1076
1077     echo -e "${optie}$argum${N}"
1078
1079     $OPT_DEBUG && log --info "$argum"
1080
1081     return 0
1082 }
1083
1084 #-Functie-----#
1085 # Naam: log           #
1086 # Doel: Tekst naar de log/journal sturen, eventueel met prioriteit info, #
1087 #       warning, of error.           #
1088 # Arg.: Twee optionele argumenten:   #
1089 #       1. prio      string, invoer:  #
1090 #       -i, --info, -w, --warning, -e, --err, --error           #
1091 #       1. tekst     string, invoer   #
1092 #-----#
1093 log() {
1094     $OPT_DEBUG || return 0
1095
1096     local optie='info'
1097     local argum=''
1098
1099     while [[ @$@ ]]; do
1100         case $1 in
1101             -i|--info)
1102                 optie='info'
1103                 shift
1104         esac
1105     done
```

```

1106         continue
1107         ;;
1108     -w|--warning)
1109         optie='warning'
1110         shift
1111         continue
1112         ;;
1113     -e|--err|--error)
1114         optie='err'
1115         shift
1116         continue
1117         ;;
1118     *) # Argument tekst.
1119         if [[ -z $argum ]]; then
1120             argum=$1
1121         else
1122             argum="$argum $1"
1123         fi
1124         shift
1125         continue
1126         ;;
1127     esac
1128 done
1129
1130 [[ $argum ]] && echo -e "$argum" |& $LOGCMD --priority $optie
1131
1132 return 0
1133 }
1134
1135 #-Functie-----#
1136 # Naam: maak_uitvoermappen_aan_sc #
1137 # Doel: Maak uitvoermappen aan die een script nodig heeft. #
1138 # Arg.: Geen argumenten. #
1139 #-----#
1140 maak_uitvoermappen_aan_sc() {
1141     log "$SCNAME:$FUNCNAME:$LINENO"
1142
1143     # Verwijder oude uitvoermappen.
1144     if [[ -d $PROGDIR/.output ]]; then
1145         rm $VERBOSE \
1146             --recursive \
1147             --force \
1148             "$PROGDIR/.output" |& $LOGCMD
1149         log "Map $HOME/.output is verwijderd."
1150     fi
1151
1152     # Maak uitvoermappen aan.
1153     if [[ ! -d $OUTDIR1 ]]; then
1154         # Niveau 1 - map - éénmalig.
1155         mkdir $VERBOSE \
1156             --parents \
1157             "$OUTDIR1" |& $LOGCMD
1158         verwerk_rc "$SCNAME:$FUNCNAME:$LINENO" \
1159             'maak uitvoermap aan (niveau 1)' \
1160             $?
1161         if $SCRIPT_NEEDS_SUDO; then
1162             chown $VERBOSE \
1163                 $SUDO_USER:$SUDO_USER \
1164                 "$OUTDIR1" |& $LOGCMD
1165             verwerk_rc "$SCNAME:$FUNCNAME:$LINENO" \
1166                 'wijzig rechten uitvoermap (niveau 1)' \
1167                 $?
1168         fi
1169     fi
1170     if [[ ! -d $OUTDIR2 ]]; then

```

```

1171     # Niveau 2 - map - éénmalig per script.
1172     mkdir      $VERBOSE \
1173               "$OUTDIR2" |& $LOGCMD
1174     verwerk_rc "$SCNAME:$FUNCNAME:$LINENO" \
1175               'maak uitvoermap aan (niveau 2)' \
1176               $?
1177     if $SCRIPT_NEEDS_SUDO; then
1178         chown   $VERBOSE \
1179               $SUDO_USER:$SUDO_USER \
1180               "$OUTDIR2" |& $LOGCMD
1181     verwerk_rc "$SCNAME:$FUNCNAME:$LINENO" \
1182               'wijzig rechten uitvoermap (niveau 2)' \
1183               $?
1184     fi
1185 fi
1186
1187 return 0
1188 }
1189
1190 #-Functie-----#
1191 # Naam: spinner #
1192 # Doel: Laat een spinner verschijnen of verdwijnen. #
1193 # Arg.: Eén verplicht argument: #
1194 #     1. spinoptie string, invoer: #
1195 #     aan spinner verschijnt en draait, of #
1196 #     uit spinner stopt en verdwijnt. #
1197 #-----#
1198 spinner() {
1199     log "$SCNAME:$FUNCNAME:$LINENO:$@"
1200
1201     local spinopt=${1:-none}
1202
1203     case $spinopt in
1204         aan)
1205             # Maak de cursor onzichtbaar.
1206             tput civis
1207
1208             # Oneindige loop. while :, while [ 1 -eq 1 ], of
1209             # while [[ 1 -eq 1 ]] werkt ook.
1210             (while true; do
1211                 echo -en '\b'
1212                 sleep $SPINNER_DELAY
1213                 echo -en '-\b'
1214                 sleep $SPINNER_DELAY
1215                 echo -en '\\\b'
1216                 sleep $SPINNER_DELAY
1217                 echo -en '| \b'
1218                 sleep $SPINNER_DELAY
1219             done) &
1220             # Bewaar spinner-proces-identificatie.
1221             SPINNER_PID=$!
1222             # Zeer korte tijd tussen spinner aan en spinner uit opvangen.
1223             sleep 0.1
1224             ;;
1225         uit)
1226             kill $SPINNER_PID |& $LOGCMD
1227             if $SCRIPT_KILLED && [[ $SPINNER_PID -gt 0 ]]; then
1228                 info "\nAchtergrondtaak spinner gestopt (proces \
1229 $SPINNER_PID)."
1230             fi
1231             # Voorkom dat er een /, -, \, of | achterblijft.
1232             echo -en ' \b'
1233             # Maak de cursor weer zichtbaar.
1234             tput cvvis
1235             ;;

```

```
1236     esac
1237
1238     return 0
1239 }
1240
1241 #-Functie-----#
1242 # Naam: success                                     #
1243 # Doel: Tekst groen tonen en naar de log/journal sturen. #
1244 # Arg.: Eén optioneel argument:                     #
1245 #     1. tekst      string, invoer                  #
1246 #-----#
1247 success() {
1248
1249     echo -e "${S}${@}${N}"
1250
1251     $OPT_DEBUG && log "$@"
1252
1253     return 0
1254 }
1255
1256 #-Functie-----#
1257 # Naam: timer                                       #
1258 # Doel: Bepaal de verstreken tijd ("elaps") als [DD-]UU:MM:SS.MMM. #
1259 # Arg.: Twee verplichte argumenten:                 #
1260 #     1. starttijd   string, invoer                  #
1261 #                   epochseconden's'milliseconden #
1262 #     1. eindtijd    string, invoer                  #
1263 #                   epochseconden's'milliseconden #
1264 #-----#
1265 timer() {
1266     log "$SCNAME:$FUNCNAME:$LINENO:$@"
1267
1268     local statime=${1:-0}
1269     local stotime=${2:-0}
1270
1271     local staepochsecs=0
1272     local stoepochsecs=0
1273
1274     local stamillisecs=0
1275     local stomillisecs=0
1276
1277     local -i diffcorr=0
1278     local -i difftime=0
1279     local -i diffdays=0
1280     local -i diffhour=0
1281     local -i diffmins=0
1282     local -i diffsecs=0
1283     local -i diffmsec=0
1284
1285     # Splits af de epochseconden en de milliseconden.
1286     staepochsecs=$(echo $statime | cut --delimiter='s' --fields=1)
1287     stoepochsecs=$(echo $stotime | cut --delimiter='s' --fields=1)
1288
1289     stamillisecs=$(echo $statime | cut --delimiter='s' --fields=2)
1290     stomillisecs=$(echo $stotime | cut --delimiter='s' --fields=2)
1291
1292     # Verwijder voorlooppullen, bash interpreteert deze als octale waarden.
1293     # vb: waarde is te groot voor basis (het onjuiste symbool is "029")
1294     # De $( ( )) zet een rekenkundige verband op en de 10# zet het getal om
1295     # van basis 10 naar basis 10 waardoor voorlooppullen verwijderd worden.
1296     stamillisecs=$(( 10#$stamillisecs ))
1297     stomillisecs=$(( 10#$stomillisecs ))
1298
1299     # Als het aantal milliseconden bij de start groter is dan aan het einde,
1300     # leen dan 1000 msec van de epochseconden en corrigeer de epochseconden.
```



```

1366     # Toon en bewaar doorlooptijd.
1367     if [[ $diffdays -eq 0 ]]; then
1368         printf "Script $PROGNAME duurde %02d:%02d:%02d.%03d.\r\n" \
1369             $diffhour $diffmins $diffsecs $diffmsec
1370         printf "Script $PROGNAME duurde de vorige keer \
1371             %02d:%02d:%02d.%03d.\r\n" \
1372             $diffhour $diffmins $diffsecs $diffmsec \
1373             > "$LASTRUN"
1374     else
1375         printf "Script $PROGNAME duurde %d-%02d:%02d:%02d.%03d.\r\n" \
1376             $diffdays $diffhour $diffmins $diffsecs $diffmsec
1377         printf "Script $PROGNAME duurde de vorige keer \
1378             %d-%02d:%02d:%02d.%03d.\r\n" \
1379             $diffdays $diffhour $diffmins $diffsecs $diffmsec \
1380             > "$LASTRUN"
1381     fi
1382
1383     return 0
1384 }
1385
1386 #-Functie-----#
1387 # Naam: toon_afgebroken_sc #
1388 # Doel: Melding naar gebruiker en log dat script is afgebroken. #
1389 # Arg.: Geen argumenten. #
1390 #------#
1391 toon_afgebroken_sc() {
1392     log "$SCNAME:$FUNCNAME:$LINENO"
1393
1394     local msg=''
1395
1396     # Stop eventuele spinner.
1397     SCRIPT_KILLED=true
1398     spinner uit
1399
1400     [[ $STATIME = 0 ]] && STATIME=$(date +%s%3N)
1401     [[ $STOTIME = 0 ]] && STOTIME=$(date +%s%3N)
1402     msg="Script $PROGNAME is afgebroken $(toon_tijd $STOTIME)."
1403
1404     warning $msg
1405
1406     warning "$(timer $STATIME $STOTIME)"
1407
1408     # Toon schermbericht (bubble) en geef "fout"-geluid.
1409     notify-send --urgency=critical \
1410         --icon=error \
1411         "De $PROGNAME is afgebroken!" \
1412         "Zie terminalvenster voor meer informatie" |& $LOGCMD
1413     canberra-gtk-play --id=suspend-error \
1414         --volume=-20 |& $LOGCMD &
1415
1416     return 0
1417 }
1418
1419 #-Functie-----#
1420 # Naam: toon_afsluiten_sc #
1421 # Doel: Melding naar gebruiker en log dat script gereed is. #
1422 # Arg.: Eén optioneel argument: #
1423 #     1. verbose boolean, invoer: #
1424 #         leeg/true #
1425 #         verwerking met output (standaard) #
1426 #         false verwerking zonder output #
1427 #------#
1428 toon_afsluiten_sc() {
1429     log "$SCNAME:$FUNCNAME:$LINENO:$@"
1430     local verbose=${1:-true}

```

```

1431
1432     $verbose || return 0
1433
1434     # Toon schermbericht (bubble) en geef "goed"-geluid.
1435     notify-send --urgency=normal \
1436                --icon=utilities-terminal \
1437                "De $PROGRAMME is uitgevoerd" \
1438                "Zie terminalvenster voor meer informatie" |& $LOGCMD
1439     canberra-gtk-play --id=complete \
1440                      --volume=-20 |& $LOGCMD &
1441
1442     return 0
1443 }
1444
1445 #-Functie-----#
1446 # Naam: toon_controleer_meldingen_sc #
1447 # Doel: Waarschuwing naar gebruiker en log. #
1448 # Arg.: Geen argumenten. #
1449 #-----#
1450 toon_controleer_meldingen_sc() {
1451     log "$SCNAME:$FUNCNAME:$LINENO"
1452
1453     warning 'Controleer de eventuele meldingen hierboven, en'
1454     warning 'controleer de log via opdracht:'
1455
1456     local epochsecs=$(echo $STATIME | cut --delimiter='s' --fields=1)
1457     local epochstime=$(date --date="@'$epochsecs '+%T')
1458
1459     command_exists systemd &&
1460     info --command "journalctl -ab -t $PROGRAMME -S $epochstime" ||
1461     info --command "grep $PROGRAMME /var/log/syslog"
1462
1463     return 0
1464 }
1465
1466 #-Functie-----#
1467 # Naam: toon_gebruik_sc #
1468 # Doel: Geeft aan hoe het script aangeroepen kan worden. #
1469 # Arg.: Geen argumenten. #
1470 #-----#
1471 toon_gebruik_sc() {
1472     log "$SCNAME:$FUNCNAME:$LINENO:$@"
1473
1474     local text=''
1475
1476     text="$PROGRAMME $OPTS_USAGE_SC$OPTS_USAGE"
1477     if $OPT_NEEDS_ARG; then
1478         text=$text"\n"
1479         text=$text"\nEen argument dat verplicht of optioneel is voor een \
1480 lange optie,"
1481         text=$text"\nis dat ook voor de overeenkomstige korte optie."
1482     fi
1483     if $OPT_USAGE; then
1484         text=$text'\n'
1485         text=$text"\nTyp '$PROGRAMME --help' voor meer informatie."
1486     else
1487         text=$text'\n'$HELP
1488     fi
1489
1490     echo -e "$text" | less $LESS_OPTIONS
1491
1492     return 0
1493 }
1494
1495 #-Functie-----#

```

```

1496 # Naam: toon_gestart_sc #
1497 # Doel: Melding naar gebruiker en log dat script is gestart. #
1498 # Arg.: Eén optioneel argument: #
1499 #     1. verbose      boolean, invoer: #
1500 #         leeg/true #
1501 #             verwerking met output (standaard) #
1502 #         false      verwerking zonder output #
1503 #-----#
1504 toon_gestart_sc() {
1505     log "$SCNAME:$FUNCNAME:$LINENO:$@"
1506     local verbose=${1:-true}
1507
1508     local first_call=false
1509     local msg=''
1510
1511     if [[ $STATIME = 0 ]]; then
1512         STATIME=$(date +%s%3N)
1513         first_call=true
1514     fi
1515
1516     $verbose && info "Script $PROGNAME is gestart $(toon_tijd $STATIME)."

```



```
1561 # Arg.: Eén verplicht argument: #
1562 #     1. tijd      string, invoer #
1563 #           (seconds since 1970-01-01 00:00:00 UTC's'milliseconds) #
1564 #-----#
1565 toon_tijd() {
1566     log "$SCNAME:$FUNCNAME:$LINENO:$@"
1567
1568     local time=${1:-0}
1569
1570     local epochsecs=$(echo $time | cut --delimiter='s' --fields=1)
1571     local millisecs=$(echo $time | cut --delimiter='s' --fields=2)
1572     echo "$(date --date=@'$epochsecs '+%Y-%m-%d') \
1573 $(date --date=@'$epochsecs '+%T').$millisecs"
1574
1575     return 0
1576 }
1577
1578 #-Functie-----#
1579 # Naam: toon_keuze_fout_sc #
1580 # Doel: Toon afbreken en breek script af. #
1581 # Arg.: Geen argumenten. #
1582 #-----#
1583 toon_keuze_fout_sc() {
1584     log "$SCNAME:$FUNCNAME:$LINENO"
1585
1586     info 'Foutieve keuze - afbreken.'
1587     exit 0
1588 }
1589
1590 #-Functie-----#
1591 # Naam: toon_keuze_stop_sc #
1592 # Doel: Toon stoppen en stop script. #
1593 # Arg.: Geen argumenten. #
1594 #-----#
1595 toon_keuze_stop_sc() {
1596     log "$SCNAME:$FUNCNAME:$LINENO"
1597
1598     info 'Stoppen.'
1599     exit_sc 0
1600 }
1601
1602 #-Functie-----#
1603 # Naam: toon_versie #
1604 # Doel: Toon versie en auteursrechten. #
1605 # Arg.: Geen argumenten. #
1606 #-----#
1607 toon_versie() {
1608     log "$SCNAME:$FUNCNAME:$LINENO"
1609
1610     info "$PROGNAME $VERSION_NUMBER"
1611     if [[ $FIRST_COPYRIGHTYEAR -eq $THIS_YEAR ]]; then
1612         copyright_years=$THIS_YEAR
1613     else
1614         copyright_years=$FIRST_COPYRIGHTYEAR-$THIS_YEAR
1615     fi
1616     info "Auteursrecht © $copyright_years Karel Zimmer."
1617     info "$GPL"
1618     info
1619     info "Geschreven door Karel Zimmer, http://karelzimmer.nl, \
1620 <info@karelzimmer.nl>."
1621
1622     return 0
1623 }
1624
1625 #-Functie-----#
```

```

1626 # Naam: verwerk_rc #
1627 # Doel: Verwerk de afsluitwaarde (returncode, rc) van een opdracht. #
1628 # Arg.: Twee verplichte argumenten: #
1629 #     1. plek string, invoer, plek waarvandaan de rc van een #
1630 #     opdracht wordt gecontroleerd, opgedeeld in 3 #
1631 #     delen gescheiden door ':' #
1632 #     2. beschrijving string, invoer, functionele beschrijving van de #
1633 #     opdracht die wordt gecontroleerd #
1634 #     3. returncode integer, invoer, de rc van de opdracht #
1635 #-----#
1636 verwerk_rc() {
1637     log "$SCNAME:$FUNCNAME:$LINENO:$@"
1638
1639     local from=${1:-$SCNAME:$FUNCNAME:$LINENO}
1640     local desc=${2:-unspecified-task}
1641     local -i rc=${3:-0}
1642
1643     [[ $rc -gt 0 ]] || return 0
1644
1645     local prog=$(echo $from | awk -F ':' '{print $1}')
1646     local func=$(echo $from | awk -F ':' '{print $2}')
1647     local line=$(echo $from | awk -F ':' '{print $3}')
1648     local rc_text=''
1649
1650     case $rc in
1651         $_COMMAND_NOT_EXECUTABLE)
1652             rc_text=' (Opdracht gevonden maar niet uitvoerbaar)'
1653             ;;
1654         $_COMMAND_NOT_FOUND)
1655             rc_text=' (Opdracht niet gevonden)'
1656             ;;
1657         $_INVALID_EXIT_ARGUMENT)
1658             rc_text=' (Ongeldig exit argument)'
1659             ;;
1660         $_INVALID_EXIT_STATUS)
1661             rc_text=' (Ongeldige exit status)'
1662             ;;
1663         *)
1664             rc_text=''
1665             ;;
1666     esac
1667
1668     error "Er is een fout opgetreden."
1669     info "    Script: $prog"
1670     info "    Functie: $func"
1671     info "    Regelnummer: $line"
1672     info "    Beschrijving: $desc"
1673     info "    Afsluitwaarde: $rc$rc_text"
1674
1675     toon_controleer_meldingen_sc
1676     toon_afgebroken_sc
1677     exit_sc $_TASK_ERROR
1678 }
1679
1680 #-Functie-----#
1681 # Naam: verwerk_signal_exit #
1682 # Doel: Handelt stoppen van het script af. #
1683 # Arg.: Eén verplicht argument: #
1684 #     1. signaalspecificatie string, invoer: #
1685 #     INT gestopt door de gebruiker met Ctrl-C (SIGINT) #
1686 #     TERM afgebroken met de kill opdracht (SIGTERM) #
1687 #     HUP afgebroken door het sluiten van het terminalvenster #
1688 #     (SIGHUP) #
1689 #     * afgebroken met een onbekend signaal #
1690 #-----#

```

```

1691 verwerk_signal_exit() {
1692     log "$SCNAME:$FUNCNAME:$LINENO:$@"
1693
1694     local sig_spec=${1:-$FUNCNAME-unspecified-signal}
1695
1696     # Stop eventuele spinner.
1697     SCRIPT_KILLED=true
1698     spinner uit
1699
1700     info
1701     case $sig_spec in
1702         INT)
1703             warning 'Gestopt door de gebruiker met Ctrl-C (SIGINT).'
1704             ;;
1705         TERM)
1706             warning 'Afgebroken met de kill opdracht (SIGTERM).'
1707             ;;
1708         HUP)
1709             warning "Afgebroken door het sluiten van het terminalvenster \
1710 (SIGHUP).\"
1711             ;;
1712         *)
1713             warning "Afgebroken met onbekend signaal $sig_spec.\"
1714             ;;
1715     esac
1716
1717     # Opruimen ("clean up") voor we stoppen.
1718     case $PROGNAME in
1719         backup)
1720             if [[ $TO_DELETE ]]; then
1721                 info 'Opgeslagen back-up verwijderen...'
1722                 rm $VERBOSE \
1723                     "$TO_DELETE" |& $LOGCMD
1724                 verwerk_rc "$SCNAME:$FUNCNAME:$LINENO" \
1725                     'verwijder opgeslagen back-up' \
1726                     $?
1727                 success 'Opgeslagen back-up is verwijderd.'
1728             fi
1729             ;;
1730         getstarted)
1731             rm $VERBOSE $PROGDIR/s |& $LOGCMD
1732             rm $VERBOSE /tmp/script-common.sh |& $LOGCMD
1733             ;;
1734     esac
1735
1736     toon_afgebroken_sc
1737     exit_sc $E_SIGNAL_EXIT
1738 }
1739
1740 #-Functie-----#
1741 # Naam: warning #
1742 # Doel: Tekst geel tonen en naar de journal sturen. #
1743 # Arg.: Eén optioneel argument: #
1744 # 1. tekst string, invoer #
1745 #-----#
1746 warning() {
1747
1748     echo -e "${W}$@${N}"
1749
1750     $OPT_DEBUG && log --warning "$@"
1751
1752     return 0
1753 }
1754
1755 return 0

```

1756

1757 # Einde script.