

```
1  #!/bin/bash
2  #####
3  # Bestand:  deploy #
4  # Doel:    Distribueer checklists, scripts, en overige bestanden. #
5  # Gebruik: In het terminalvenster: #
6  #         deploy [OPTIES] #
7  #         Zie ook script getclean en script getstarted. #
8  # Gebruikt: Uitvoerbestanden (in ~/.output/deploy/): #
9  #         deploy.rpt      (rapport van verschillen) #
10 #         deploy.diff     (verschillen tussen uploads en uploads.old) #
11 #         deploy.diff.tmp1 (verschillen met aangepaste bestandsnamen) #
12 #         deploy.diff.tmp2 (verschillen gesorteerd en geünificeerd) #
13 # Auteur:   Karel Zimmer (http://karelzimmer.nl, info@karelzimmer.nl) #
14 # ----- #
15 # Auteursrecht © 2009-2017 Karel Zimmer. #
16 # #
17 # Dit programma is vrije software: u mag het herdistribueren en/of wijzigen #
18 # onder de voorwaarden van de GNU Algemene Publieke Licentie zoals #
19 # gepubliceerd door de Free Software Foundation, onder versie 3 van de #
20 # Licentie of (naar uw keuze) elke latere versie. #
21 # #
22 # Dit programma is gedistribueerd in de hoop dat het nuttig zal zijn maar #
23 # ZONDER ENIGE GARANTIE; zelfs zonder de impliciete garanties die #
24 # GEBRUIKELIJK ZIJN IN DE HANDEL of voor BRUIKBAARHEID VOOR EEN SPECIFIEK #
25 # DOEL. Zie de GNU Algemene Publieke Licentie voor meer details. #
26 # #
27 # U hoort een kopie van de GNU Algemene Publieke Licentie te hebben #
28 # ontvangen samen met dit programma. Als dat niet het geval is, zie #
29 # http://www.gnu.org/licenses/. #
30 # ----- #
31 # Versies:  1.0.0    2009-02-14  Eerste versie. #
32 #           2.0.0    2009-05-13  Meer berichten naar de gebruiker. #
33 #           3.0.0    2011-01-28  Maak shellarchiefbestand aan. #
34 #           4.0.0    2012-02-08  Archiefmapnaamgeving gewijzigd (+ <JJJJ>). #
35 #           5.0.0    2012-08-10  Archiefmapnaam inclusief scriptnaam. #
36 #           6.0.0    2012-10-06  Optie notodo toegevoegd, structuur uploads #
37 #                               gewijzigd (uploads/data en uploads/site). #
38 #           7.0.0    2012-10-27  Gewijzigd aankoppelpunt en archiefmap. #
39 #           8.0.0    2012-11-30  Kopieer ook checklists (.odt) naar uploads. #
40 #           9.0.0    2012-12-08  Rapporteer verschillen, geen todo meer. #
41 #          10.0.0    2013-01-23  Kopieer ook Windows-scripts (.cmd). #
42 #          11.0.0    2013-02-10  Optie check toegevoegd. #
43 #          12.0.0    2013-03-03  Optie print en md5sum-controle toegevoegd. #
44 #          13.0.0    2013-03-18  Opties waar nodig aanzetten of uitzetten. #
45 #          14.0.0    2013-06-24  Optie edit toegevoegd. #
46 #          15.0.0    2013-10-22  Optie check verwijderd. #
47 #          16.0.0    2013-11-09  Kopieer geen Windows-scripts (.cmd) meer. #
48 #          17.0.0    2014-01-01  Optie ignore toegevoegd. #
49 #          18.0.0    2014-01-26  Opslag script-PDF vanuit gedit gewijzigd. #
50 #          19.0.0    2014-05-03  Verwerk desktopbestanden (.desktop). #
51 #          20.0.0    2015-01-28  Verwerk checklists en tekstbestanden. #
52 #          21.0.0    2015-01-29  Controle op computernaam en synchroniseer #
53 #                               site naar Google Drive. #
54 #          22.0.0    2015-03-11  Synchroniseer scans naar Google Drive. #
55 #          23.0.0    2015-04-01  Synchroniseer naar Google Drive gelijk USB. #
56 #          24.0.0    2015-04-04  Wacht op synchronisatie naar Google Drive. #
57 #          25.0.0    2015-04-05  Optie ignore verwijderd. #
58 #          26.0.0    2015-05-01  Optie force verwijderd. #
59 #          27.0.0    2015-07-08  Hernoemd (-/-.sh) en sync ook met ~/bin. #
60 #          28.0.0    2015-09-06  Sync met ~/bin verwijderd, doet getscripts. #
61 #          29.0.0    2016-05-20  Van eigen log naar system log en systemd #
62 #                               journal. #
63 #          30.0.0    2016-06-16  Van MD5 naar SHA256. #
64 #          32.0.0    2016-09-21  Optie scripts toegevoegd. #
65 #          33.0.0    2016-11-14  Optie checklists toegevoegd. #
```

```
66 #          34.0.0   2017-05-07  Optie list toegevoegd.          #
67 #          35.0.0   2017-05-10  Optie force toegevoegd.         #
68 #          36.0.0   2017-06-06  Ook scripts.arch/dev en uploads deployen. #
69 #####
70 readonly VERSION_NUMBER=36.4.1
71 readonly RELEASE_DATE=2017-08-22
72
73 #####
74 # Instellingen          #
75 #####
76
77 #-----#
78 # Algemene instellingen          #
79 #-----#
80 readonly FIRST_COPYRIGHTYEAR=2009      # Eerste auteursrechtjaar
81 readonly PROGDIR=$(cd "$(dirname "${BASH_SOURCE[0]}")" && pwd)
82                                     # Programmamap
83 readonly PROGNAME=$(basename "${BASH_SOURCE[0]}")
84                                     # Programmaam
85 readonly SCRIPT_NEEDS_SUDO=false      # Beheerdersrechten nodig
86 if ! source "$PROGDIR"/script-common.sh; then
87     echo "
88 Het algemeen scriptbestand (script-common.sh) is niet gevonden of bevat \
89 fouten.
90 Is 'wget karelzimmer.nl/s;. s' wel uitgevoerd?
91 Zie voor meer informatie http://karelzimmer.nl, en klik op LEESMIJ." >&2
92 # Return als gesourced (0=bash of -bash).
93     [[ $0 == *bash ]] && return 1 || exit 1
94 fi                                  # Lees alg. variabelen en functies in
95
96 #-----#
97 # Globale constanten          #
98 #-----#
99 readonly OUTDIR=$HOME/.output/$PROGNAME # Uitvoermap
100                                     # Afbeeldingenmap en afgeleiden:
101 readonly AFBDIR=$XDG_PICTURES_DIR
102 readonly AFBDIRSCN=$AFBDIR/Scans
103 readonly AFBDIRWPP=$AFBDIR/wallpapers.d
104                                     # Documentenmap en afgeleiden:
105 readonly DOCDIR=$XDG_DOCUMENTS_DIR
106 readonly DOCDIRCL=$DOCDIR/Checklists
107 readonly DOCDIRCLBASE=$(basename $DOCDIRCL)
108 readonly DOCDIRADR=$DOCDIR/Adressen
109 readonly DOCDIRKOV=$DOCDIR/Overig
110
111 readonly MUSICDIR=$XDG_MUSIC_DIR      # Muziekmap
112
113 readonly PGMDIR="$HOME/Programma's"  # Programmamap
114                                     # Scriptsmap en afgeleiden:
115 readonly SCRIPTSDIR=$HOME/scripts
116 readonly SCRIPTSDIRBASE=$(basename $SCRIPTSDIR)
117 readonly SCRIPTSARCHDIR=$SCRIPTSDIR.arch
118 readonly SCRIPTSARCHDIRBASE=$SCRIPTSDIRBASE.arch
119 readonly SCRIPTSARCHDIRYR=$SCRIPTSARCHDIR/$THIS_YEAR
120 readonly SCRIPTSARCHDIRYRBASE=$SCRIPTSARCHDIRBASE/$THIS_YEAR
121 readonly SCRIPTSDEVDIR=$SCRIPTSDIR.dev
122 readonly SCRIPTSDEVDIRBASE=$(basename $SCRIPTSDEVDIR)
123                                     # Uploadsmap en afgeleiden:
124 readonly UPLOADSDIR=$HOME/uploads
125 readonly UPLOADSDIRBASE=$(basename $UPLOADSDIR)
126 readonly UPLOADSDIRDAT=$UPLOADSDIR/data
127 readonly UPLOADSDIRDATLUX=$UPLOADSDIRDAT/linux
128 readonly UPLOADSDIRSIT=$UPLOADSDIRBASE/site
129 readonly UPLOADSDIRWPP=$UPLOADSDIR/downloads/wallpapers
130 readonly UPLOADSDIROLD=$UPLOADSDIR.old
```

```
131
132 readonly VIDEODIR=$XDG_VIDEOS_DIR      # Videomap
133
134 readonly ADRCZ=$DOCDIRADR/Adressen_Carl_Zimmer.pdf
135 readonly                # Adreslijst Carl Zimmer
136 readonly ADRHH=$DOCDIRADR/adreslijst_Hoonhout.pdf
137 readonly                # Adreslijst Hoonhout
138 readonly KPXKDB=$DOCDIRKOV/Sleutelbos_Karel.kdbx
139 readonly                # Sleutelbos
140
141 readonly HASHPROG=sha256sum            # Controleprogramma
142
143 readonly TARFILE_WP=wallpapers.tar     # Wallpapersarchiefbestand
144 readonly HASHFILE_WP=$TARFILE_WP.$HASHPROG
145 readonly                # Wallpaperscontrolebestand
146 readonly TARFILE_CL=checklists.tar.gz # Checklistsarchiefbestand
147 readonly HASHFILE_CL=$TARFILE_CL.$HASHPROG
148 readonly                # Checklistscontrolebestand
149 readonly TARFILE_SH=scripts.tar.gz    # Scriptsarchiefbestand
150 readonly HASHFILE_SH=$TARFILE_SH.$HASHPROG
151 readonly                # Scriptscontrolebestand
152 readonly TARFILE_SA=scripts.arch.tar.gz # Scriptsarchiefbestand
153 readonly HASHFILE_SA=$TARFILE_SA.$HASHPROG
154 readonly                # Scriptsarchiefcontrolebestand
155 readonly TARFILE_SD=scripts.dev.tar.gz # Ontwikkelscriptsarchiefbestand
156 readonly HASHFILE_SD=$TARFILE_SD.$HASHPROG
157 readonly                # Ontwikkelscriptscontrolebestand
158 readonly TARFILE_UP=uploads.tar.gz    # Uploadsarchiefbestand
159 readonly HASHFILE_UP=$TARFILE_UP.$HASHPROG
160 readonly                # Uploadscontrolebestand
161
162 readonly DIFFFILE=$PROGNAME.diff      # Verschillenbestand
163 readonly DIFF=$OUTDIR/$DIFFFILE      # Verschillen
164
165 readonly REPORTFILE=$PROGNAME.rpt     # Rapportbestand
166 readonly REPORT=$OUTDIR/$REPORTFILE  # Rapport
167
168 declare MOUNTDIR=/media/$USER        # Koppelpunt vanaf Ubuntu 12.10
169 if [[ ! -d $MOUNTDIR ]]; then
170     MOUNTDIR=/media                  # Koppelpunt voor Ubuntu 12.10
171 fi
172 readonly MOUNTDIR                    # Koppelpunt verwisselbare media
173
174 readonly USBSTICKDIR=$MOUNTDIR/SanDisk128G
175 readonly                # USB-stick-mappen
176 readonly USBSTICK_SYNCDIR=$USBSTICKDIR/A
177 readonly                # USB-stick-syncmap (i.v.m. Live USB)
178 readonly GRIVE_SYNCDIR="$HOME/Google Drive/A"
179 readonly                # Google Drive-syncmap
180 readonly TO_SYNC_USB="\
181     $AFBDIR
182     $DOCDIR
183     $MUSICDIR
184     $PGMDIR
185     $SCRIPTSDIR
186     $SCRIPTSARCHDIR
187     $SCRIPTSDEVDIR
188     $UPLOADSDIR
189     $UPLOADSDIROLD
190     $VIDEODIR"
191 readonly TO_SYNC_GD="\
192     $AFBDIRSCN
193     $ADRCZ
194     $ADRHH
195     $KPXKDB"
readonly                # Synchroniseer met Google Drive
```



```
261 # Array, integer =====
262
263 # Boolean =====
264 declare OPT_CHECKLISTS=false # Optie shecklists opgegeven
265 declare OPT_EDIT=false # Optie edit opgegeven
266 declare OPT_FORCE=false # Optie force opgegeven
267 declare OPT_LIST=false # Optie list opgegeven
268 declare OPT_PRINT=false # Optie print opgegeven
269 declare OPT_RESET=false # Optie reset opgegeven
270 declare OPT_SCRIPTS=false # Optie scripts opgegeven
271 declare USBSTICK=false # USB-stick aanwezig
272
273 # Tekst =====
274
275 #####
276 # Functies #
277 #####
278
279 #-Functie-----#
280 # Naam: controleer_invoer #
281 # Doel: Controleer en verwerk de invoer van het script. #
282 # Arg.: Script invoer. #
283 #-----#
284 controleer_invoer() {
285
286     local dir=''
287     local parm_error=false
288
289     # Test op aanwezigheid nieuwe versie getopt.
290     set +o errexit
291     getopt --test > /dev/null
292     getopt_rc=$?
293     set -o errexit
294     if [[ $getopt_rc -eq 4 ]]; then
295
296         # Ontleed (parse) opties en argumenten met getopt.
297         PARSED=$(getopt --alternative \
298             --options $OPTS_SHORT \
299             --longoptions $OPTS_LONG \
300             --name "$PROGNAME" \
301             -- "$@")
302
303         # Controleer op fouten.
304         if [[ $? -gt 0 ]]; then
305             info $HELPLINE
306             exit 2
307         fi
308
309         # Gebruik eval om quoting correct te verwerken.
310         eval set -- "$PARSED"
311     else
312         # Voeg '--' toe aan de invoer i.v.m. eindeloze loop (break op '--').
313         eval set -- "$@ --"
314         log "Oude versie van getopt in gebruik; opties exact opgeven \
315 volgens optie usage (-u)"
316     fi
317
318     # Verwerk de opties en optie-argumenten.
319     while true; do
320         case $1 in
321             -c|--checklists)
322                 OPT_CHECKLISTS=true
323                 shift
324                 ;;
325             -e|--edit)
```

```
326         OPT_EDIT=true
327         shift
328         ;;
329     -f|--force)
330         OPT_FORCE=true
331         shift
332         ;;
333     -l|--list)
334         OPT_LIST=true
335         shift
336         ;;
337     -p|--print)
338         OPT_PRINT=true
339         shift
340         ;;
341     -r|--reset)
342         OPT_RESET=true
343         shift
344         ;;
345     -s|--scripts)
346         OPT_SCRIPTS=true
347         shift
348         ;;
349     -d|--debug)
350         OPT_DEBUG=true
351         VERBOSE='--verbose'
352         shift
353         ;;
354     -h|--help)
355         OPT_HELP=true
356         toon_gebruik $OPT_USAGE
357         exit 0
358         ;;
359     -u|--usage)
360         OPT_USAGE=true
361         toon_gebruik $OPT_USAGE
362         exit 0
363         ;;
364     -v|--version)
365         OPT_VERSION=true
366         toon_versie
367         exit 0
368         ;;
369     --)
370         shift
371         break
372         ;;
373     *)
374         error "$PROGNAME: interne fout bij afhandelen optie '$1'"
375         parm_error=true
376         shift
377         ;;
378     esac
379 done
380
381 # Verwerk de non-optie argumenten.
382 if [[ $# ]; then
383     info "$PROGNAME: geen argumenten toegestaan"
384     parm_error=true
385 fi
386
387 # Geef optie help en gebruik aan als er fouten zijn geconstateerd.
388 if $parm_error; then
389     info $HELPLINE
390     exit 3
```

```
391     fi
392
393     # Controleer computernaam.
394     if [[ $HOSTNAME != $DEVMACH ]]; then
395         info "$PROGNAME: alleen uit te voeren op $DEVMACH"
396         exit 0
397     fi
398
399     # Controleer aanwezigheid "hard-coded" bronmappen.
400     for dir in "$AFBDIR" \
401               "$AFBDIRWPP" \
402               "$DOCDIR" \
403               "$DOCDIRCL" \
404               "$DOCDIRADR" \
405               "$DOCDIRKOV" \
406               "$SCRIPTSDIR" \
407               "$SCRIPTSARCHDIR" \
408               "$SCRIPTSDEVDIR" \
409               "$UPLOADSDIR" \
410               "$UPLOADSDIRDAT" \
411               "$UPLOADSDIRDATLUX" \
412               "$UPLOADSDIRWPP" \
413               "$UPLOADSDIROLD"; do
414         if [[ ! -d $dir ]]; then
415             error "$PROGNAME: map '$dir' bestaat niet"
416             exit $E_DIR_NOT_FOUND
417         fi
418     done
419
420     # Controleer de starters, installatie-, en instellingsbestanden voor
421     # gebruik door deze gebruiker.
422     if $(grep --quiet \
423           --exclude='deploy' \
424           --exclude='getstarted' \
425           --regex='/home/karel/bin/' \
426           $SCRIPTSDIR/* \
427           2> /dev/null); then
428         error "$PROGNAME: bestanden in $SCRIPTSDIR bevatten \
429 '/home/karel/bin/'"
430         info "Script getstarted moet '/home/karel/scripts/' kunnen wijzigen
431 naar '/home/$USER/bin/'."
432 Anders kunnen scripts niet via de starters (.desktop) en
433 installatie- en instellingsbestanden (.sh) worden gestart."
434         exit $E_BINDIR_FOUND
435     fi
436
437     # Maak de map archiefmap/<jaartal> aan.
438     if [[ ! -d $SCRIPTSARCHDIRYR ]]; then
439         mkdir $VERBOSE \
440              "$SCRIPTSARCHDIRYR" |& $LOGCMD
441     fi
442
443     # Maak de Google Drive-map aan.
444     if [[ ! -d $GRIVE_SYNC_DIR ]]; then
445         mkdir $VERBOSE \
446              "$GRIVE_SYNC_DIR" |& $LOGCMD
447     fi
448
449     $OPT_LIST && toon_wijzigingen
450
451     $OPT_PRINT && print_wijzigingen
452
453     wijzig_verschillenbestand $OPT_EDIT $OPT_RESET
454
455     $OPT_FORCE || controleer_usbstick
```

```
456
457     return 0
458 }
459
460 #-Functie-----#
461 # Naam: toon_wijzigingen #
462 # Doel: Toon het overzicht met wijzigingen. #
463 # Arg.: Geen argumenten. #
464 #-----#
465 toon_wijzigingen() {
466
467     if [[ -s "$REPORT" ]]; then
468         cat "$REPORT"
469     else
470         info "$PROGNAME: rapportbestand is leeg"
471     fi
472     exit 0
473
474     return 0
475 }
476
477 #-Functie-----#
478 # Naam: print_wijzigingen #
479 # Doel: Druk op verzoek van de gebruiker het overzicht wijzigingen af. #
480 # Arg.: Eén verplicht argument: #
481 # Arg.: Geen argumenten. #
482 #-----#
483 print_wijzigingen() {
484
485     if [[ -s "$REPORT" ]]; then
486         cat "$REPORT"
487         warning "Overzicht wijzigingen wordt afgedrukt."
488         read -p 'Doorgaan met afdrukken? (j/N) '
489         case $REPLY in
490             j*|J*)
491                 lpr "$REPORT" |& $LOGCMD
492                 info 'Overzicht wijzigingen naar printer gestuurd.'
493                 ;;
494             n*|N*|'')
495                 info 'Overzicht wijzigingen wordt niet afgedrukt.'
496                 ;;
497             *)
498                 info 'Foutieve keuze - afbreken.'
499                 exit 0
500                 ;;
501         esac
502     else
503         info "$PROGNAME: rapportbestand is leeg"
504     fi
505     exit 0
506
507     return 0
508 }
509
510 #-Functie-----#
511 # Naam: wijzig_verschillenbestand #
512 # Doel: Wijzig het zogenaamde verschillenbestand met vi of maak deze leeg. #
513 # Arg.: Twee verplichte argumenten: #
514 #     1. edit-optie boolean, invoer: #
515 #         true verschillenbestand wordt geopend met vi #
516 #         false verschillenbestand wordt niet geopend met vi #
517 #     2. reset-optie boolean, invoer: #
518 #         true verschillenbestand wordt na bevestiging leeggemaakt #
519 #         false verschillenbestand wordt niet leeggemaakt #
520 #-----#
```



```
521 wijzig_verschillenbestand() {
522
523     local editopt=${1:-false}
524     local resetopt=${2:-false}
525
526     # Controleer opties.
527     if $editopt && $resetopt; then
528         info "$PROGNAME: specificieer optie reset of optie edit, niet allebei"
529         exit $E_MUTUALLY_EXCLUSIVE_OPTS
530     fi
531
532     # Verwerk optie edit.
533     if $editopt; then
534         if [[ -s "$DIFF" ]]; then
535             vi "$DIFF"
536             info 'Verschillenbestand gewijzigd.'
537         else
538             info "$PROGNAME: verschillenbestand is leeg"
539         fi
540         read -p 'Doorgaan met deploy? (j/N) '
541         case $REPLY in
542             j*|J*)
543             log 'Gebruiker kiest doorgaan met deploy.'
544             ;;
545             n*|N*|'')
546             info 'Stoppen.'
547             exit 0
548             ;;
549             *)
550             info 'Foutieve keuze - afbreken.'
551             exit 0
552             ;;
553         esac
554     fi
555
556     # Verwerk optie reset.
557     if $resetopt; then
558         if [[ -s "$DIFF" ]]; then
559             info "$PROGNAME: verschillenbestand wordt leeggemaakt"
560             read -p 'Doorgaan met leegmaken? (j/N) '
561             case $REPLY in
562                 j*|J*)
563                 rm $VERBOSE \
564                     "$DIFF" \
565                     "$REPORT" |& $LOGCMD
566                 info "Verschillenbestand is leeggemaakt."
567                 ;;
568                 n*|N*|'')
569                 info "Verschillenbestand is niet leeggemaakt."
570                 ;;
571                 *)
572                 info 'Foutieve keuze - afbreken.'
573                 exit 0
574                 ;;
575             esac
576         else
577             rm $VERBOSE \
578                 "$REPORT" |& $LOGCMD
579             info "$PROGNAME: verschillenbestand is al leeg"
580         fi
581         read -p 'Doorgaan met deploy? (j/N) '
582         case $REPLY in
583             j*|J*)
584                 log 'Gebruiker kiest doorgaan met deploy.'
585             ;;
```

```

586         n*|N*|''')
587         info 'Stoppen.'
588         exit 0
589         ;;
590     *)
591         info 'Foutieve keuze - afbreken.'
592         exit 0
593         ;;
594     esac
595 fi
596
597 return 0
598 }
599
600 #-Functie-----#
601 # Naam: controleer_usbstick #
602 # Doel: Controleer aanwezigheid USB-stick en syncmap. #
603 # Arg.: Geen argumenten. #
604 #-----#
605 controleer_usbstick() {
606
607     # Controleer aanwezigheid USB-stick.
608     if [[ -d $USBSTICKDIR ]]; then
609         USBSTICK=true
610     else
611         info "$PROGNAME: USB-stick "${basename "$USBSTICKDIR"}" is niet \
612 aanwezig"
613         read -p 'Negeren en doorgaan met deploy? (j/N) '
614         case $REPLY in
615             j*|J*)
616             log 'USB-stick ontbreekt, gebruiker kiest doorgaan.'
617             ;;
618             n*|N*|''')
619             info 'Stoppen.'
620             exit 0
621             ;;
622             *)
623             info 'Foutieve keuze - afbreken.'
624             exit 0
625             ;;
626         esac
627     fi
628
629     # Maak indien nodig de USB-stick-map aan.
630     if $USBSTICK && [[ ! -d $USBSTICK_SYNCNCDIR ]]; then
631         mkdir $VERBOSE \
632             "$USBSTICK_SYNCNCDIR" |& $LOGCMD
633     fi
634
635     return 0
636 }
637
638 #-Functie-----#
639 # Naam: toon_invoer #
640 # Doel: Toon wat het script gaat doen. #
641 # Arg.: Geen argumenten. #
642 #-----#
643 toon_invoer() {
644
645     info "          Script: $PROGNAME
646          --- Deploy VAN ---
647     Afbeeldingen: $AFBDIR
648     Documenten: $DOCDIR
649     Muziek: $MUSICDIR
650     Programma's: $PGMDIR

```

```
651         Scripts: $SCRIPTSDIR
652         Video's: $VIDEODIR
653         --- Deploy NAAR ---
654     Scripts-archief: $SCRIPTSARCHDIRYR
655     Google Drive: $GRIVE_SYNC_DIR
656                 http://drive.google.com/ (internet)
657     Uploads: $UPLOADSDIR"
658     if $USBSTICK; then
659         info "         USB-stick: $USBSTICK_SYNC_DIR"
660     else
661         warning '         USB-stick: Ontbreekt'
662     fi
663     info "
664     Rapport: $REPORT
665 "
666
667     return 0
668 }
669
670 #-Functie-----#
671 # Naam: verwerk_invoer                                #
672 # Doel: Synchroniseer mappen en bestanden, en print wijzigingen. #
673 # Arg.: Geen argumenten.                               #
674 #------#
675 verwerk_invoer() {
676
677     maak_uitvoermappen_aan
678     verwijder_backupbestanden
679     archiveer_scriptbestanden
680     synchroniseer_uploads
681     maak_scriptsarchief_aan
682     maak_scriptsarchiefarchief_aan
683     maak_ontwikkelscriptsarchief_aan
684     maak_checklistsarchief_aan
685     maak_uploadsarchief_aan
686     maak_wallpapersarchief_aan
687     success 'FileZilla mag nu gebruikt worden om te uploaden.'
688     attendeer_gebruiker ''
689     bepaal_wijzigingen
690     synchroniseer_uploads_oud
691     synchroniseer_google_drive
692     verwerk_usbstick
693
694     return 0
695 }
696
697 #-Functie-----#
698 # Naam: maak_uitvoermappen_aan                        #
699 # Doel: Maak uitvoermappen aan die het script nodig heeft. #
700 # Arg.: Geen argumenten.                               #
701 #------#
702 maak_uitvoermappen_aan() {
703
704     # Verwijder oude uitvoermappen.
705     info
706     if [[ -d $PROGDIR/.output ]]; then
707         rm $VERBOSE \
708             --recursive \
709             --force \
710             "$PROGDIR/.output" |& $LOGCMD
711     fi
712
713     # Maak uitvoermap aan.
714     if [[ ! -d $OUTDIR ]]; then
715         mkdir $VERBOSE \
```

```

716             --parents \
717             "$OUTDIR"  |& $LOGCMD
718     fi
719
720     return 0
721 }
722
723 #-Functie-----#
724 # Naam: verwijder_backupbestanden #
725 # Doel: Verwijder back-upbestand zoals gedit die aanmaakt. #
726 # Arg.: Geen argumenten. #
727 #-----#
728 verwijder_backupbestanden() {
729     info 'Verwijder back-upbestanden...'
730
731     find $TO_SYNC_USB \
732         -name '*~' -delete -or \
733         -name '*.*~' -delete -or \
734         -name '*.bak' -delete |& $LOGCMD
735
736     return 0
737 }
738
739 #-Functie-----#
740 # Naam: archiveer_scriptbestanden #
741 # Doel: Archiveer de scriptbestanden (indien mogelijk met toevoeging van #
742 # versienummer en -datum). #
743 # Arg.: Geen argumenten. #
744 #-----#
745 archiveer_scriptbestanden() {
746
747     local archivefile=''
748     local file=''
749     local -i num_files=0
750
751     info 'Archiveer scriptbestanden...'
752
753     # Ga naar scriptsmap.
754     cd "$SCRIPTSDIR" \
755         2> /dev/null
756
757     # Archiveer scriptbestanden.
758     for file in *; do
759
760         # Verwerk alleen reguliere bestanden, geen map o.i.d.
761         [[ ! -f $file ]] && continue
762
763         controleer_archiefmap "$SCRIPTSARCHDIRYR/$file"
764         bepaal_archiefbestandsnaam "$SCRIPTSDIR" \
765             "$file" \
766             archivefile
767
768         # Meld welk bestand.
769         log "$file -> $archivefile"
770
771         # Naar archief.
772         cp --update \
773            --preserve \
774            --recursive \
775            $VERBOSE \
776            "$SCRIPTSDIR/$file" \
777            "$SCRIPTSARCHDIRYR/$file/$archivefile" |& $LOGCMD
778         chmod $VERBOSE \
779            a-x \
780            "$SCRIPTSARCHDIRYR/$file/$archivefile" |& $LOGCMD

```

```

781     num_files=$(( num_files + 1 ))
782     done
783     info "Er zijn $num_files scriptbestanden gearchiveerd."
784
785     return 0
786 }
787
788 #-Functie-----#
789 # Naam: controleer_archiefmap                                     #
790 # Doel: Maak indien nodig de archiefmap aan.                   #
791 # Arg.: Eén verplicht argument:                                #
792 #     1. mapnaam      string, invoer                            #
793 #-----#
794 controleer_archiefmap() {
795
796     local archivedir=${1:-/tmp}
797
798     if [[ ! -d $archivedir ]]; then
799         mkdir          $VERBOSE          \
800             "$archivedir"      |& $LOGCMD
801     fi
802
803     return 0
804 }
805
806 #-Functie-----#
807 # Naam: bepaal_archiefbestandsnaam                             #
808 # Doel: Stelt een archiefbestandsnaam samen (voor het archief). #
809 # Arg.: Drie verplichte argumenten:                            #
810 #     1. scriptmapnaam      string, invoer                      #
811 #     2. scriptbestandsnaam string, invoer                      #
812 #     3. variabelenaam      string, uitvoer, krijgt als inhoud  #
813 #     archiefbestandsnaam                                     #
814 #-----#
815 bepaal_archiefbestandsnaam() {
816
817     local scriptdir=${1:-/tmp}
818     local scriptfile=${2:-file}
819     local _archivefile_variable=${3:-archivefile}
820     local _archivefile_value=''
821     local filename=''
822     local filesuffix=''
823     local release_date=''
824     local version_number=''
825
826     # Zoek met grep 'VERSION_NUMBER=' en 'RELEASE_DATE=' in te kopiëren
827     # bestand en splits versienummer en versiedatum af met cut. Grep met
828     # max-count=1 vindt alleen de eerste bovenin het bestand. Bijvoorbeeld
829     # VERSION_NUMBER=1.1.0 en RELEASE_DATE=2009-06-20 in backup.sh levert
830     # respectievelijk 1.1.0 en 2009-06-20.
831     version_number=$(
832         grep      --no-filename          \
833                 --max-count=1          \
834                 'VERSION_NUMBER='      \
835                 "$scriptdir/$scriptfile" |
836         cut      --delimiter='='        \
837                 --fields=2             |
838         awk      '{print $1}'           || true
839     )
840     release_date=$(
841         grep      --no-filename          \
842                 --max-count=1          \
843                 'RELEASE_DATE='        \
844                 "$scriptdir/$scriptfile" |
845         cut      --delimiter='='        \

```

```

846         --fields=2           |
847         awk '{print $1}'     || true
848     )
849
850     # Controleer auteursrechtjaar.
851     copyright_year=$(
852         grep --fixed-strings \
853             --no-filename \
854             --max-count=1 \
855             'FIRST_COPYRIGHTYEAR=' \
856             "$scriptdir/$scriptfile" |
857         cut --delimiter='=' \
858            --fields=2           |
859         awk '{print $1}'     || true
860     )
861
862     if [[ $copyright_year -gt $THIS_YEAR ]]; then
863         error "$scriptfile: eerste auteursrechtjaar '$copyright_year' ligt \
864 voor dit jaar '$THIS_YEAR'"
865         exit $E_INVALID_DATE
866     fi
867
868     # Controleer vrijgavedatum.
869     if ! date --date="$release_date" &> /dev/null; then
870         error "$scriptfile: vrijgavedatum '$release_date' is niet geldig"
871         info "Pas de vrijgavedatum in bestand '$scriptfile' aan en start \
872 dit script opnieuw."
873         exit $E_INVALID_DATE
874     fi
875     local today=$(date +%Y%m%d)
876     local release=$(
877         date +%Y%m%d \
878         --date=$release_date
879     )
880     if [[ $release -gt $today ]]; then
881         error "$scriptfile: vrijgavedatum '$release_date' ligt in de \
882 toekomst"
883         info "Pas de vrijgavedatum in bestand '$scriptfile' aan en start \
884 dit script opnieuw."
885         exit $E_INVALID_DATE
886     fi
887     # Bijvoorbeeld, precies een jaar oud:
888     # today =20161012
889     # release=20151012
890     # age = 10000
891     local age=$(( $today - $release ))
892     local to_old=$(date +%Y-%m-%d --date='1 year ago')
893     if [[ $age -gt 10000 ]]; then
894         warning "$scriptfile: vrijgavedatum '$release_date' is meer dan een \
895 jaar geleden"
896     fi
897
898     # Archiefbestandsnaam is afhankelijk aanwezigheid versienummer en
899     # vrijgavedatum.
900     if [[ -z $version_number ]]; then
901
902         # Geen 'VERSION_NUMBER=' gevonden, kopieer naar Archief "as-is".
903         _archivefile_value=$scriptfile
904     else
905
906         # Bepaal bestandsnaam en suffix. Bijvoorbeeld backup.sh levert als
907         # bestandsnaam backup en als suffix sh, README.txt levert README en
908         # txt, en test alleen test (geen suffix).
909         case $scriptfile in
910             *.* )

```

```

911         filename=${scriptfile%.*}
912         filesuffix=${scriptfile##*.}
913         ;;
914     *)
915         filename=$scriptfile
916         filesuffix=''
917         ;;
918     esac
919
920     # Kopieer naar Archief met toevoeging versienummer en -datum na de
921     # bestandsnaam en voor de suffix. Bijvoorbeeld backup.sh als
922     # backup-2009-06-20-1.1.0.sh en test als test-2009-06-20.
923     _archivefile_value=$filename-$release_date-$version_number$filesuffix
924 fi
925
926 eval $_archivefile_variable="$_archivefile_value"
927
928 return 0
929 }
930
931 #-Functie-----#
932 # Naam: synchroniseer_uploads #
933 # Doel: Synchroniseer de checklists, scripts, installatie- en instellings- #
934 # bestanden (voor de site en script getstarted) met uploads. #
935 # Arg.: Geen argumenten. #
936 #-----#
937 synchroniseer_uploads() {
938     info 'Synchroniseer uploads...'
939
940     # Verplaats Checklists-PDFs en scripts-PDFs naar de juiste plek.
941     verplaats_pdfs
942
943     # Synchroniseer Checklists en scripts met uploadsmap.
944     rsync --archive \
945         --delete \
946         --exclude='.*' \
947         --exclude='*.pdf' \
948         $VERBOSE \
949         "$DOCDIRCL"/ \
950         "$SCRIPTSDIR"/ \
951         "$UPLOADSDIRDATLUX" |& $LOGCMD
952
953     return 0
954 }
955
956 #-Functie-----#
957 # Naam: verplaats_pdfs #
958 # Doel: Verplaats PDFs ontstaan bij een export vanuit LibreOffice en print #
959 # naar bestand vanuit gedit. #
960 # Arg.: Geen argumenten. #
961 #-----#
962 verplaats_pdfs() {
963
964     # Verplaats scripts-PDFs (uit gedit) naar uploads.
965     if ls "$DOCDIR"/*.pdf &> /dev/null; then
966         mv $VERBOSE \
967             "$DOCDIR"/*.pdf \
968             "$UPLOADSDIRDATLUX"/ |& $LOGCMD
969     fi
970
971     # Verplaats checklist-PDFs (uit LibreOffice) naar uploads.
972     if ls "$DOCDIRCL"/*.pdf &> /dev/null; then
973         mv $VERBOSE \
974             "$DOCDIRCL"/*.pdf \
975             "$UPLOADSDIRDATLUX"/ |& $LOGCMD

```

```
976     fi
977
978     return 0
979 }
980
981 #-Functie-----#
982 # Naam: maak_scriptsarchief_aan #
983 # Doel: Maak archief aan van alles in de scriptsmap inclusief een #
984 #       controlebestand en zet deze in de uploadmap. #
985 #       Doe dit vanuit de scriptsmap zelf en niet vanuit ~ i.v.m. ~/bin. #
986 # Arg.: Geen argumenten. #
987 #-----#
988 maak_scriptsarchief_aan() {
989     info 'Maak scriptsarchief aan...'
990
991     # Ga naar scriptsmap.
992     cd "$SCRIPTSDIR"
993
994     # Verwijder eventueel aanwezig archief en controlebestand.
995     rm $VERBOSE \
996        "$TARFILE_SH" \
997        "$HASHFILE_SH" |& $LOGCMD || true
998
999     # Maak archief.
1000    tar --create \
1001        --gzip \
1002        $VERBOSE \
1003        * \
1004        .icons/* \
1005        --file="$TARFILE_SH" |& $LOGCMD
1006
1007    # Maak controlebestand.
1008    $HASHPROG --text \
1009        "$TARFILE_SH" \
1010        > "$HASHFILE_SH" |& $LOGCMD
1011
1012    # Verplaats archief en controlebestand naar de uploadmap, tenzij optie
1013    # scripts is opgegeven.
1014    if ! $OPT_SCRIPTS; then
1015        mv $VERBOSE \
1016           "$TARFILE_SH" \
1017           "$HASHFILE_SH" \
1018           "$UPLOADSDIRDATLUX" |& $LOGCMD
1019    fi
1020
1021    return 0
1022 }
1023
1024 #-Functie-----#
1025 # Naam: maak_scriptsarchiefarchief_aan #
1026 # Doel: Maak archief aan van dit jaar in de scriptsarchiefmap inclusief een #
1027 #       controlebestand en zet deze in de uploadmap. #
1028 # Arg.: Geen argumenten. #
1029 #-----#
1030 maak_scriptsarchiefarchief_aan() {
1031
1032     info 'Maak scriptsarchiefarchief aan...'
1033
1034     # Ga naar thuismap.
1035     cd "$HOME"
1036
1037     # Verwijder eventueel aanwezig archief en controlebestand.
1038     rm $VERBOSE \
1039        "$TARFILE_SA" \
1040        "$HASHFILE_SA" |& $LOGCMD || true
```



```

1041
1042 # Maak archief.
1043 tar --create \
1044 --gzip \
1045 $VERBOSE \
1046 "$SCRIPTSARCHDIRYRBASE" \
1047 --file="$TARFILE_SA" |& $LOGCMD
1048
1049 # Maak controlebestand.
1050 $HASHPROG --text \
1051 "$TARFILE_SA" \
1052 > "$HASHFILE_SA" |& $LOGCMD
1053
1054 # Verplaats archief en controlebestand naar de uploadmap.
1055 mv $VERBOSE \
1056 "$TARFILE_SA" \
1057 "$HASHFILE_SA" \
1058 "$UPLOADSDIRDATLUX" |& $LOGCMD
1059
1060 return 0
1061 }
1062
1063 #-Functie-----#
1064 # Naam: maak_ontwikkelscriptsarchief_aan #
1065 # Doel: Maak archief aan van ontwikkelscriptsmap inclusief een #
1066 # controlebestand en zet deze in de uploadmap. #
1067 # Arg.: Geen argumenten. #
1068 #-#
1069 maak_ontwikkelscriptsarchief_aan() {
1070
1071 info 'Maak ontwikkelscriptsarchief aan...'
1072
1073 # Ga naar thuismap.
1074 cd "$HOME"
1075
1076 # Verwijder eventueel aanwezig archief en controlebestand.
1077 rm $VERBOSE \
1078 "$TARFILE_SD" \
1079 "$HASHFILE_SD" |& $LOGCMD || true
1080
1081 # Maak archief.
1082 tar --create \
1083 --gzip \
1084 $VERBOSE \
1085 "$SCRIPTSDEVDIRBASE" \
1086 --file="$TARFILE_SD" |& $LOGCMD
1087
1088 # Maak controlebestand.
1089 $HASHPROG --text \
1090 "$TARFILE_SD" \
1091 > "$HASHFILE_SD" |& $LOGCMD
1092
1093 # Verplaats archief en controlebestand naar de uploadmap.
1094 mv $VERBOSE \
1095 "$TARFILE_SD" \
1096 "$HASHFILE_SD" \
1097 "$UPLOADSDIRDATLUX" |& $LOGCMD
1098
1099 return 0
1100 }
1101
1102 #-Functie-----#
1103 # Naam: maak_checklistsarchief_aan #
1104 # Doel: Maak archief aan van alles in de checklistsmap inclusief een #
1105 # controlebestand en zet deze in de uploadmap. #

```

```
1106 #       Doe dit vanuit de XDG_DOCUMENTS_DIR en niet vanuit ~ i.v.m. l10n. #
1107 # Arg.: Geen argumenten. #
1108 #-----#
1109 maak_checklistsarchief_aan() {
1110     info 'Maak checklistsarchief aan...'
1111     # Ga naar documentenmap.
1112     cd "$XDG_DOCUMENTS_DIR"
1113     # Verwijder eventueel aanwezig archief en controlebestand.
1114     rm $VERBOSE \
1115        "$STARFILE_CL" \
1116        "$HASHFILE_CL" |& $LOGCMD || true
1117     # Maak archief aan.
1118     tar --create \
1119         --gzip \
1120         $VERBOSE \
1121         "$DOCDIRCLBASE"/*.odt \
1122         "$DOCDIRCLBASE"/*.txt \
1123         --file="$STARFILE_CL" |& $LOGCMD
1124     # Maak controlebestand aan.
1125     $HASHPROG --text \
1126        "$STARFILE_CL" \
1127        > "$HASHFILE_CL" |& $LOGCMD
1128     # Verplaats archief en controlebestand naar de uploadmap, tenzij optie
1129     # checklists is opgegeven.
1130     if ! $OPT_CHECKLISTS; then
1131         mv $VERBOSE \
1132            "$STARFILE_CL" \
1133            "$HASHFILE_CL" \
1134            "$UPLOADSDIRDATLUX" |& $LOGCMD
1135     fi
1136     return 0
1137 }
1138
1139 #-Functie-----#
1140 # Naam: maak_uploadsarchief_aan #
1141 # Doel: Maak archief aan van site in de uploadmap inclusief een #
1142 #       controlebestand en zet deze in de uploadmap. #
1143 # Arg.: Geen argumenten. #
1144 #-----#
1145 maak_uploadsarchief_aan() {
1146     info 'Maak uploadsarchief aan...'
1147     # Ga naar thuismap.
1148     cd "$HOME"
1149     # Verwijder eventueel aanwezig archief en controlebestand.
1150     rm $VERBOSE \
1151        "$STARFILE_UP" \
1152        "$HASHFILE_UP" |& $LOGCMD || true
1153     # Maak archief.
1154     tar --create \
1155         --gzip \
1156         $VERBOSE \
1157         "$UPLOADSDIRSIT" \
1158         --file="$STARFILE_UP" |& $LOGCMD
1159 }
```

```
1171     # Maak controlebestand.
1172     $HASHPROG --text \
1173     "$STARFILE_UP" \
1174     > "$HASHFILE_UP" |& $LOGCMD
1175
1176     # Verplaats archief en controlebestand naar de uploadmap.
1177     mv $VERBOSE \
1178     "$STARFILE_UP" \
1179     "$HASHFILE_UP" \
1180     "$UPLOADSDIRDATLUX" |& $LOGCMD
1181
1182     return 0
1183 }
1184
1185 #-Functie-----#
1186 # Naam: maak_wallpapersarchief_aan #
1187 # Doel: Maak archief aan van alles in de wallpapersmap inclusief een #
1188 #       controlebestand en zet deze in de uploadmap. #
1189 #       Doe dit vanuit de wallpapersmap zelf en niet vanuit ~ i.v.m. ~/opt. #
1190 # Arg.: Geen argumenten. #
1191 #-----#
1192 maak_wallpapersarchief_aan() {
1193
1194     info 'Maak wallpapersarchief aan...'
1195
1196     # Ga naar wallpapersmap.
1197     cd "$AFBDIRWPP"
1198
1199     # Verwijder eventueel aanwezig archief en controlebestand.
1200     rm $VERBOSE \
1201     "$STARFILE_WP" \
1202     "$HASHFILE_WP" |& $LOGCMD || true
1203
1204     # Maak archief.
1205     tar --create \
1206     $VERBOSE \
1207     * \
1208     --file="$STARFILE_WP" |& $LOGCMD
1209
1210     # Maak controlebestand.
1211     $HASHPROG --text \
1212     "$STARFILE_WP" \
1213     > "$HASHFILE_WP" |& $LOGCMD
1214
1215     # Verplaats archief en controlebestand naar de uploadmap.
1216     mv $VERBOSE \
1217     "$STARFILE_WP" \
1218     "$HASHFILE_WP" \
1219     "$UPLOADSDIRWPP" |& $LOGCMD
1220
1221     return 0
1222 }
1223
1224 #-Functie-----#
1225 # Naam: bepaal_wijzigingen #
1226 # Doel: Bepaal en toon verschillen (wijzigingen) tussen uploads en #
1227 #       uploads-oud. #
1228 # Arg.: Geen argumenten. #
1229 #-----#
1230 bepaal_wijzigingen() {
1231     local hdr=''
1232     local msg=''
1233     local record=''
1234     local recordtmp=''
1235     local -i num_of_changes=0
```

```
1236
1237 # Bepaal verschillen (toevoegen aan DIFF).
1238 info 'Bepaal wijzigingen...'
1239
1240 diff --brief \
1241      --recursive \
1242      --exclude="$TARFILE_SH" \
1243      --exclude="$TARFILE_SA" \
1244      --exclude="$TARFILE_SD" \
1245      --exclude="$TARFILE_CL" \
1246      --exclude="$TARFILE_UP" \
1247      --exclude="$HASHFILE_SH" \
1248      --exclude="$HASHFILE_SA" \
1249      --exclude="$HASHFILE_SD" \
1250      --exclude="$HASHFILE_CL" \
1251      --exclude="$HASHFILE_UP" \
1252      "$UPLOADSDIR" \
1253      "$UPLOADSDIROLD" \
1254      >> "$DIFF" || true
1255
1256 # Voorkom fout als optie reset en als er geen verschillen zijn.
1257 touch "$DIFF.tmp1"
1258
1259 # Wijzig verschillenlijst (van DIFF naar DIFF.tmp1).
1260 # -----
1261 # Verwijder aan het begin 'Bestanden ' en alles vanaf ' en /'. Verwijder
1262 # aan het begin 'Alleen in' en vervang '<pad>: <bestand>' door
1263 # '<pad>/<bestand>'. Rapporteer (met basenamen) alleen de bestandsnaam
1264 # bij .odt en .pdf, voorbeeld:
1265 # - Checklist Archiveren.odt = Checklist Archiveren
1266 # Checklist Archiveren.pdf = Checklist Archiveren
1267 # - install-Ubuntu-12.10-PC01.sh = install-Ubuntu-12.10-PC01.sh
1268 # install-Ubuntu-12.10-PC01.sh.pdf = install-Ubuntu-12.10-PC01.sh
1269 # - info.txt = info.txt
1270 # info.txt.pdf = info.txt
1271 # - backup = backup
1272 # backup.pdf = backup
1273 while read record; do
1274     recordtmp=$record
1275
1276     recordtmp=${recordtmp#Bestanden }
1277     recordtmp=${recordtmp% en /*}
1278
1279     recordtmp=${recordtmp#Alleen in }
1280     recordtmp=$(echo "$recordtmp" | sed 's/: /\//g')
1281
1282     recordtmp=$(basename "$recordtmp" .odt)
1283     recordtmp=$(basename "$recordtmp" .pdf)
1284
1285     echo "__ $recordtmp" >> "$DIFF.tmp1"
1286 done < "$DIFF"
1287
1288 # Sorteert en uniceert verschillenrecords (van DIFF.tmp1 naar DIFF.tmp2).
1289 sort --unique "$DIFF.tmp1" >> "$DIFF.tmp2"
1290
1291 # Maak en toon verschillenbestand met pr (uit DIFF.tmp2 naar REPORT).
1292 msg='--- Overzicht wijzigingen ---'
1293 info "BEGIN $msg"
1294 num_of_changes=$(( $(wc --lines "$DIFF.tmp2" | awk '{print $1}') ))
1295 hdr='--- Overzicht wijzigingen ($num_of_changes) ---'
1296 pr --indent=6 \
1297    --date-format='%a %d %b %Y' \
1298    --form-feed \
1299    --header="$hdr" \
1300    "$DIFF.tmp2" |
```

```
1301     tee "$REPORT"
1302     $OPT_DEBUG && cat "$REPORT" |& $LOGCMD
1303     info "EINDE $msg"
1304
1305     # Verwijder tijdelijke bestanden (DIFF.tmp<n>).
1306     rm $VERBOSE \
1307        "$DIFF".tmp* |& $LOGCMD
1308
1309     return 0
1310 }
1311
1312 #-Functie-----#
1313 # Naam: synchroniseer_uploads_oud #
1314 # Doel: Synchroniseer uploads met uploads-oud (om wijzigingen te kunnen #
1315 #       bepalen). #
1316 # Arg.: Geen argumenten. #
1317 #-----#
1318 synchroniseer_uploads_oud() {
1319
1320     info 'Synchroniseer uploads-oud...'
1321
1322     rsync      --archive      \
1323              --delete       \
1324              $VERBOSE       \
1325              "$UPLOADSDIR"/  \
1326              "$UPLOADSDIROLD" |& $LOGCMD
1327
1328     return 0
1329 }
1330
1331 #-Functie-----#
1332 # Naam: synchroniseer_google_drive #
1333 # Doel: Synchroniseer met Google Drive. #
1334 # Arg.: Geen argumenten. #
1335 #-----#
1336 synchroniseer_google_drive() {
1337     info 'Synchroniseer Google Drive...'
1338
1339     # Synchroniseer naar Google Drive. exclude in verband met (verborgen)
1340     # configuratiebestanden, inplace, size-only, en whole-file in verband met
1341     # google-drive-ocamlfuse.
1342     rsync      --delete      \
1343              --inplace      \
1344              --exclude=.*   \
1345              --recursive    \
1346              --size-only    \
1347              $VERBOSE       \
1348              --whole-file   \
1349              $TO_SYNC_GD    \
1350              "$GRIVE_SYNC_DIR" |& $LOGCMD
1351
1352     return 0
1353 }
1354
1355 #-Functie-----#
1356 # Naam: verwerk_usbstick #
1357 # Doel: Synchroniseer USB-stick (indien van toepassing). #
1358 # Arg.: Geen argumenten. #
1359 #-----#
1360 verwerk_usbstick() {
1361     local device=''
1362     local -i unmount_rc=0
1363
1364     if $USBSTICK; then
1365         synchroniseer_usbstick
```

```
1366     info 'Ontkoppel USB-stick...'
1367     # umount/unmount via udisksctl (udisks command line tool).
1368     device=$(
1369         findmnt "$USBSTICKDIR" \
1370             --list \
1371             --noheadings \
1372             --output=SOURCE)
1373     udisksctl unmount \
1374         --block-device $device |& $LOGCMD
1375     unmount_rc=$?
1376     udisksctl power-off \
1377         --block-device $device |& $LOGCMD
1378     if [[ $unmount_rc -eq 0 ]]; then
1379         success 'USB-stick is ontkoppeld en kan verwijderd worden.'
1380     else
1381         warning 'USB-stick zelf ontkoppelen en verwijderen.'
1382     fi
1383 else
1384     warning "Geen USB-stick geplaatst.
1385 Voer script $PROGNAME later nogmaals uit met een geplaatste USB-stick."
1386 fi
1387
1388     return 0
1389 }
1390
1391 #-Functie-----#
1392 # Naam: synchroniseer_usbstick #
1393 # Doel: Synchroniseer de USB-stick met het systeem. #
1394 # Arg.: Geen argumenten. #
1395 #-----#
1396 synchroniseer_usbstick() {
1397     info 'Synchroniseer USB-stick...'
1398
1399     # Synchroniseer naar USB-stick.
1400     #-----#
1401     # --no-perms in verband met het doel FAT-filesystem,
1402     # --modify-window in verband met het FAT-filesystem,
1403     # --exclude in verband met (verborgen) configuratiebestanden.
1404     rsync --archive \
1405         --delete \
1406         --no-perms \
1407         --modify-window=1 \
1408         $VERBOSE \
1409         $TO_SYNC_USB \
1410         "$USBSTICK_SYNCDIR" |& $LOGCMD
1411
1412     # Laat weten wanneer de laatste deploy op USB-stick is uitgevoerd.
1413     rm $VERBOSE \
1414         "$USBSTICK_SYNCDIR/last-deploy-*" |& $LOGCMD
1415     echo "Timestamp: $TIMESTAMP" > \
1416         "$USBSTICK_SYNCDIR/last-deploy-$TIMESTAMP"
1417
1418     return 0
1419 }
1420
1421 #-Functie-----#
1422 # Naam: toon_afsluiten #
1423 # Doel: Afsluitende meldingen naar gebruiker en log. #
1424 # Arg.: Geen argumenten. #
1425 #-----#
1426 toon_afsluiten() {
1427
1428     info "
1429 Voer de volgende akties uit:
1430 1. Controleer --- Overzicht wijzigingen ---"
```

```
1431 2. Controleer bij site-wijzigingen
1432     1. de HTML en CSS via ${C}ckhtml${N}
1433     2. de koppelingen via ${B}LinkChecker${N}
1434 3. Voer de IVP uit met ${C}ivpsh${N}, en als IVP succesvol:
1435     1. Werk karelzimmer.nl bij met ${B}FileZilla${N}
1436     2. Werk ~/bin bij met ${C}getstarted${N} of
1437         ${C}wget karelzimmer.nl/s;. s${N}
1438 4. Commit de wijzigingen op https://GitHub.com."
1439
1440     # Geef "goed"-geluid en (als non-root) toon schermbericht ("bubble").
1441     command_exists canberra-gtk-play &&
1442         canberra-gtk-play --id=complete |& $LOGCMD
1443     command_exists notify-send &&
1444         notify-send --urgency=normal \
1445             --icon=utilities-terminal \
1446             "De $PROGNAME is uitgevoerd" \
1447             "Zie terminalvenster voor meer informatie" |& $LOGCMD
1448
1449     return 0
1450 }
1451
1452 #####
1453 # Hoofdlijn #
1454 #####
1455
1456 # init_script
1457 {
1458     controleer_invoer "$@"
1459     controleer_gebruiker $SCRIPT_NEEDS_SUDO "$@"
1460
1461     # Bij optie scripts alleen het scriptsarchief en het controlebestand
1462     # aanmaken en stoppen.
1463     if $OPT_SCRIPTS; then
1464         maak_scriptsarchief_aan
1465         info "$PROGNAME: $STARFILE_SH en $HASHFILE_SH aangemaakt in \
1466 $SCRIPTSDIR"
1467         exit 0
1468     fi
1469
1470     # Bij optie checklists alleen het checklistsarchief en het
1471     # controlebestand aanmaken en stoppen.
1472     if $OPT_CHECKLISTS; then
1473         maak_checklistsarchief_aan
1474         info "$PROGNAME: $STARFILE_CL en $HASHFILE_CL aangemaakt in $DOCDIRCL"
1475         exit 0
1476     fi
1477
1478     toon_invoer
1479 }
1480
1481 # verwerk
1482 {
1483     toon_gestart
1484     verwerk_invoer
1485     toon_gestopt
1486 }
1487
1488 # afsl_script
1489 {
1490     toon_afsluiten
1491     exit 0
1492 }
1493
1494 # Einde script
```