

```
1: #!/bin/bash
2: # shellcheck source=kz-common.sh
3: #####
4: # Apps instellen.
5: #
6: # Geschreven door Karel Zimmer <info@karelzimmer.nl>.
7: #####
8: PROGRAM_PATH=$(realpath "$(dirname "$0")")
9: source "$PROGRAM_PATH"/kz-common.sh
10: PROGRAM_NAME=kz-setup
11: DISPLAY_NAME=${PROGRAM_NAME}/kz-/kz }
12: RELEASE_YEAR=2013
13:
14: VERSION_NUMBER=45.05.02
15: VERSION_DATE=2021-10-09
16:
17:
18: #####
19: # Global constants
20: #####
21:
22: readonly RUN_AS_SUPERUSER=false
23: readonly OPTIONS_SHORT=${OPTIONS_SHORT_COMMON}'ace:filrs'
24: readonly OPTIONS_LONG=${OPTIONS_LONG_COMMON}",apps,cat,edition:,files,info,list,\
25: reset,simulate"
26: readonly USAGE="Gebruik: $DISPLAY_NAME [-a|--apps] [-c|--cat] \
27: [-e|--edition=EDITIE] [-f|--files]
28:             [-i|--info] [-l|--list] [-r|--reset] [-s|--simulate]
29:             $OPTIONS_USAGE_COMMON
30:             [APP...] [BESTAND...]"
31: readonly HELP="Gebruik: $DISPLAY_NAME [OPTIE...] [APP...] [BESTAND...]"
32:
33: Apps instellen.
34:
35: Opties:
36:   -a --apps      toon lijst van in te stellen apps
37:   -c --cat       toon inhoud van instellingsbestanden
38:   -e --edition=EDITIE
39:                 gebruik opgegeven editie
40:   -f --files     toon lijst van beschikbare instellingsbestanden
41:   -i --info      toon aanvullende instructies na het instellen van apps
42:   -l --list      maak instellijst aan
43:   -r --reset     toon instructies voor het resetten van apps
44:   -s --simulate  geen actie, simuleer wijzigingen, en toon opdracht die
45:                 uitgevoerd zou worden
46: $OPTIONS_HELP_COMMON
47:
48: Argumenten:
49:   APP           opgegeven apps instellen
50:   BESTAND       opgegeven instellingsbestanden verwerken"
51:
52: readonly EDITION_DEFAULT=desktop
53: DASHES=$(printf '%.0s=' {1..79}); readonly DASHES
54: SPACES=$(printf '%.0s ' {1..79}); readonly SPACES
55: VERSION=$(lsb_release --release --short); readonly VERSION
56: CMDS_FILE=$(mktemp -t "$PROGRAM_NAME-XXXXXXXXXX.cmds"); readonly CMDS_FILE
57: TEXT_FILE=$(mktemp -t "$PROGRAM_NAME-XXXXXXXXXX.text"); readonly TEXT_FILE
58:
59: # Less-opties, overgenomen (en aangepast, zie 'man less', zoek PROMPTS)
60: # van:
61: # 1. systemctl en journalctl, zie bijv. 'man systemctl', zoek LESS
62: #   ("FRSXMK")
63: # 2. man, zie 'man man', zoek LESS
64: readonly LESS_OPTIONS="--LONG-PROMPT --no-init --quit-if-one-screen \
65: --quit-on-intr --RAW-CONTROL-CHARS --prompt=M Tekstuitvoer $PROGRAM_NAME \
66: ?ltregel %lt?L/%L.:byte %bB?s/%s..?e (EINDE) :?pB %pB%. .(druk h voor \
67: hulp of q voor stoppen)"
68:
```

```
69:
70: #####
71: # Global variables
72: #####
73:
74: declare -a APP_ARGUMENT=()
75: declare -a FILE_ARGUMENT=()
76: declare -a FILE_TO_PROCESS=()
77: declare ARGUMENT_APP=false
78: declare ARGUMENT_FILE=false
79: declare COMMANDS_FOUND=false
80: declare EDITION=$EDITION_DEFAULT
81: declare EDITION_ARGUMENT=''
82: declare EXECUTE_COMMANDS=true
83: declare -i MAXRC=0
84: declare INFO_FOUND=false
85: declare OPTION_APPS=false
86: declare OPTION_CAT=false
87: declare OPTION_EDITION=false
88: declare OPTION_FILES=false
89: declare OPTION_INFO=false
90: declare OPTION_LIST=false
91: declare OPTION_RESET=false
92: declare OPTION_SIMULATE=false
93: declare RESET_FOUND=false
94:
95: # Terminalattributen, zie 'man terminfo'. Gebruik ${<variabele-naam>}.
96: declare CARRIAGE_RETURN=''
97: declare BLINK=''
98: declare CURSOR_INVISIBLE=''
99: declare CURSOR_VISIBLE=''
100:
101:
102: #####
103: # Functions
104: #####
105:
106: check_input() {
107:     local -i app_arg_num=0
108:     local -i file_arg_num=0
109:     local -i getopt_rc=0
110:     local parsed=''
111:
112:     parsed=$(
113:         getopt --alternative \
114:             --options "$OPTIONS_SHORT" \
115:             --longoptions "$OPTIONS_LONG" \
116:             --name "$DISPLAY_NAME" \
117:             -- "$@"
118:     ) || getopt_rc=$?
119:     if [[ $getopt_rc -ne $SUCCESS ]]; then
120:         printf '%s\n' "$USAGELINE" >&2
121:         exit $ERROR
122:     fi
123:     eval set -- "$parsed"
124:     process_common_options "$@"
125:     STATUS_ERROR="[${RED}]FOUT[${NORMAL}]"
126:     STATUS_SUCCESS="[${GREEN}]GOED[${NORMAL}]"
127:
128:     while true; do
129:         case $1 in
130:             -a|--apps)
131:                 OPTION_APPS=true
132:                 shift
133:                 ;;
134:             -c|--cat)
135:                 OPTION_CAT=true
136:                 shift
```

```

137:         ;;
138:     -e|--edition)
139:         if $OPTION_EDITION; then
140:             TEXT="optie '$1' eenmaal opgeven"
141:             printf "$DISPLAY_NAME: %s\n%s\n" \
142:                 "$TEXT" \
143:                 "$USAGELINE" >&2
144:             exit $ERROR
145:         else
146:             OPTION_EDITION=true
147:             EDITION_ARGUMENT=$2
148:             EDITION=$EDITION_ARGUMENT
149:         fi
150:         shift 2
151:         ;;
152:     -f|--files)
153:         OPTION_FILES=true
154:         shift
155:         ;;
156:     -i|--info)
157:         OPTION_INFO=true
158:         EXECUTE_COMMANDS=false
159:         shift
160:         ;;
161:     -l|--list)
162:         OPTION_LIST=true
163:         EXECUTE_COMMANDS=false
164:         shift
165:         ;;
166:     -r|--reset)
167:         OPTION_RESET=true
168:         EXECUTE_COMMANDS=false
169:         shift
170:         ;;
171:     -s|--simulate)
172:         OPTION_SIMULATE=true
173:         EXECUTE_COMMANDS=false
174:         shift
175:         ;;
176:     --)
177:         shift
178:         break
179:         ;;
180:     *)
181:         shift
182:         ;;
183:     esac
184: done
185:
186: if ! $OPTION_GUI && [[ -t 1 ]]; then
187:     # Tekstuitvoer non-gui naar de terminal.
188:     CARRIAGE_RETURN=$(tput cr)
189:     BLINK=$(tput blink)
190:     CURSOR_INVISIBLE=$(tput civis)
191:     CURSOR_VISIBLE=$(tput cvvis)
192: fi
193: STATUS_BUSY=" [${BLINK}${BOLD}WERK${NORMAL}]"
194:
195: # Verwerk argumenten.
196: while [[ "$*" ]]; do
197:     if [[ "$(basename "$1")" = $PROGRAM_NAME-*.sh ]]; then
198:         ARGUMENT_FILE=true
199:         FILE_ARGUMENT[$file_arg_num]=$1
200:         ((++file_arg_num))
201:         shift
202:     else
203:         ARGUMENT_APP=true
204:         APP_ARGUMENT[$app_arg_num]=$1

```

```
205:         ((++app_arg_num))
206:         shift
207:     fi
208: done
209: }
210:
211:
212: process_input() {
213:     local file=''
214:     local file_notfound=false
215:     local -i app_seq_num=0
216:     local -i app_tot_num=0
217:     local -i file_num=0
218:     local skip_line=false
219:     local dfltfile_1=''
220:     local dfltfile_2=''
221:
222:     dfltfile_1=$PROGRAM_PATH/$PROGRAM_NAME-$DISTRO-$VERSION-$EDITION.sh
223:     dfltfile_2=$PROGRAM_PATH/$PROGRAM_NAME-$DISTRO-$VERSION-$EDITION-\
224: $HOSTNAME.sh
225:     if $OPTION_LIST; then
226:         check_user
227:         process_option_list
228:         exit $SUCCESS
229:     fi
230:     if $ARGUMENT_APP; then
231:         process_argument_app
232:     fi
233:     if $ARGUMENT_FILE; then
234:         process_argument_file
235:     fi
236:
237:     # Als argument APP of FILE is opgegeven, geen standaardbestanden verwerken.
238:     if ! ($ARGUMENT_APP || $ARGUMENT_FILE); then
239:         process_dfltfiles
240:     fi
241:
242:     # Verwerk opties.
243:     if $OPTION_APPS; then
244:         process_option_apps
245:         exit $SUCCESS
246:     elif $OPTION_FILES; then
247:         process_option_files
248:         exit $SUCCESS
249:     fi
250:
251:     # Ergens hiervoor ging het niet helemaal goed.
252:     if $file_notfound; then
253:         exit $SUCCESS
254:     fi
255:
256:     # Optie cat is afhankelijk van argumenten APP en FILE.
257:     if $OPTION_CAT; then
258:         process_option_cat
259:         exit $SUCCESS
260:     fi
261:
262:     # We gaan opdrachten uitvoeren, mogen we dat wel?
263:     if $EXECUTE_COMMANDS; then
264:         check_user
265:     fi
266:
267:     # We gaan opdrachten uitvoeren of simuleren.
268:     for file in "${FILE_TO_PROCESS[@]}; do
269:         cat "$file" >> "$CMDS_FILE"
270:     done
271:
272:     # 'if ! grep --regexp='^#2'...; then...' werkt niet; grep geeft 0
```

```
273: # terug als niets gevonden, is OK, maar dan ook rc=1, is niet OK.
274: app_tot_num=$(
275:     grep --word-regexp --regexp='^#2' --count "$CMDS_FILE" || true
276: )
277:
278: if $EXECUTE_COMMANDS && $OPTION_GUI; then
279:     TITLE='Apps instellen'
280:     TEXT=''
281:     # Met "|& zenity --progress" worden globale variabelen uit
282:     # aangeroepen functies niet doorgegeven, vandaar de
283:     # 'process substitution' met "> >(zenity...)".
284:     process_file > >(
285:         zenity --progress \
286:             --pulsate \
287:             --auto-close \
288:             --no-cancel \
289:             --width 700 \
290:             --height 50 \
291:             --title "$TITLE" \
292:             --text "$TEXT" 2> >($LOGCMD)
293:     )
294: else
295:     process_file
296: fi
297: }
298:
299:
300: process_option_list() {
301:     local list_dir="$HOME/$PROGRAM_NAME"
302:     local text0='Opdrachten worden verwerkt'
303:     local text1=' [1/4] Gebruikersfoto vastleggen'
304:     local file1="$list_dir/1. $PROGRAM_NAME-gebruikersfoto"
305:     local text2=' [2/4] Bureaubladachtergrond vastleggen'
306:     local file2="$list_dir/2. $PROGRAM_NAME-bureaubladachtergrond"
307:     local text3=' [3/4] Favorieten vastleggen'
308:     local file3="$list_dir/3. $PROGRAM_NAME-favorieten"
309:     local text4=' [4/4] Opdrachtgeschiedenis vastleggen'
310:     local file4="$list_dir/4. $PROGRAM_NAME-opdrachtgeschiedenis"
311:
312:     rm --force --recursive "$list_dir"
313:     mkdir --parents "$list_dir"
314:
315:     TITLE='Instellijst maken'
316:     if $OPTION_GUI; then
317:         (
318:             printf '%s\n' "$text1"
319:             save_user_photo #1
320:             printf '%s\n' "$text2"
321:             save_desktop_background #2
322:             printf '%s\n' "$text3"
323:             save_favorite_apps #3
324:             printf '%s\n' "$text4"
325:             save_command_history #4
326:         ) |&
327:         zenity --progress \
328:             --pulsate \
329:             --auto-close \
330:             --no-cancel \
331:             --width 600 \
332:             --height 50 \
333:             --title "$TITLE" \
334:             --text "$text0" 2> >($LOGCMD)
335:     else
336:         save_user_photo #1
337:         save_desktop_background #2
338:         save_favorite_apps #3
339:         save_command_history #4
340:     fi
```

```
341:
342:     info "
343: De instellingen zijn vastgelegd in bestanden in de Persoonlijke map -> \
344: $PROGRAM_NAME.
345: Deze bestanden bevatten de eerder gebruikte:
346: 1. gebruikersfoto,
347: 2. bureaubladachtergrond,
348: 3. favorieten, en
349: 4. opdrachtgeschiedenis.
350: Deze bestanden kunnen worden gebruikt voor het nalopen van het instellen."
351: }
352:
353:
354: save_user_photo() {                #1
355:     printf '%s\n' "$text1..."
356:     if ! cp /var/lib/AccountsService/icons/"$USER" "$file1" |& $LOGCMD; then
357:         rm --force "$file1"
358:         touch "$file1"
359:     fi
360: }
361:
362:
363: save_desktop_background() {        #2
364:     printf '%s\n' "$text2..."
365:     gsettings get org.gnome.desktop.background \
366:         picture-uri > "$file2" |& $LOGCMD
367: }
368:
369:
370: save_favorite_apps() {             #3
371:     printf '%s\n' "$text3..."
372:     gsettings get org.gnome.shell \
373:         favorite-apps > "$file3" |& $LOGCMD
374: }
375:
376:
377: save_command_history() {           #4
378:     printf '%s\n' "$text4..."
379:     HISTFILE=~/.bash_history
380:     set -o history
381:     history > "$file4" |& $LOGCMD
382: }
383:
384:
385: process_argument_app() {
386:     local app=''
387:     local file=''
388:     local cmds_file=''
389:     local -a file_to_process=()
390:
391:     if [[ -e $dfلتfile_1 ]]; then
392:         file_to_process[0]=$dfلتfile_1
393:     else
394:         info "Geen installatiebestanden gevonden voor $DISTRO $VERSION \
395: $EDITION."
396:         exit $WARNING
397:     fi
398:     # Niet verplicht aanwezig.
399:     if [[ -e $dfلتfile_2 ]]; then
400:         file_to_process[1]=$dfلتfile_2
401:     fi
402:
403:     cmds_file=$(mktemp -t "$PROGRAM_NAME-XXXXXXXXXX.cmds")
404:     # Install zoekt naar apps in alle kz-install-<distro>-<versie>-bestanden
405:     # en stopt bij de eerstgevonden (installatie is computer-onafhankelijk),
406:     # setup zoekt naar apps in kz-install-<distro>-<versie>-<edition>.sh en
407:     # kz-install-<distro>-<versie>-<edition>-<computernaam>.sh.
408:     for app in "${APP_ARGUMENT[@]"; do
```

```
409:     if ! grep --files-with-matches \
410:             --line-regexp \
411:             --regexp="#1 $app" \
412:             "${file_to_process[@]}" \
413:             &> /dev/null; then
414:         info "App '$app' niet gevonden."
415:         file_notfound=true
416:     else
417:         process_argument_app_file
418:     fi
419: done
420: if $file_notfound; then
421:     info "Typ '$DISPLAY_NAME --apps' voor beschikbare apps."
422: else
423:     if ! grep --quiet --line-regexp --regexp='^# EOF' "$cmds_file"; then
424:         printf '%s\n' '# EOF' >> "$cmds_file"
425:     fi
426:     FILE_TO_PROCESS[$file_num]=$cmds_file
427:     ((++file_num))
428: fi
429: }
430:
431:
432: process_argument_app_file() {
433:     local app_found=false
434:     local description_found=false
435:     local record=''
436:     local temp_file=''
437:
438:     temp_file=$(mktemp -t "$PROGRAM_NAME-XXXXXXXXXX.temp")
439:     cat "${file_to_process[@]}" > "$temp_file"
440:
441:     while read -r record; do
442:         case $record in
443:             '#1 '* )
444:                 if [[ $record = '#1 '$app ]]; then
445:                     # Gezochte app-naam.
446:                     # Regels doorschrijven naar tijdelijk bestand tot
447:                     # eerstvolgende app-naam of beschrijving (verplicht),
448:                     # of tot EOF.
449:                     app_found=true
450:                     description_found=false
451:                 fi
452:                 ;;
453:             '#2'* )
454:                 # Beschrijving.
455:                 if $description_found; then
456:                     # Volgende beschrijving.
457:                     app_found=false
458:                     continue
459:                 elif $app_found; then
460:                     printf '%b\n' "$record" >> "$cmds_file"
461:                     description_found=true
462:                 fi
463:                 ;;
464:             *)
465:                 # Overige regels.
466:                 if $app_found; then
467:                     printf '%s\n' "$record" >> "$cmds_file"
468:                 fi
469:                 ;;
470:         esac
471:     done < "$temp_file"
472:
473:     rm "$temp_file" |& $LOGCMD
474: }
475:
476:
```

```
477: process_argument_file() {
478:     local file=''
479:
480:     for file in "${FILE_ARGUMENT[@]}; do
481:         if [[ -e $file ]]; then
482:             FILE_TO_PROCESS[$file_num]=$file
483:             ((++file_num))
484:         else
485:             info "Bestand $file niet gevonden."
486:             file_notfound=true
487:         fi
488:     done
489:     if $file_notfound; then
490:         info "Typ '$DISPLAY_NAME --files' voor beschikbare \
491: instellingsbestanden."
492:     fi
493: }
494:
495:
496: process_dfltfiles() {
497:     if [[ -e $dfltfile_1 ]]; then
498:         FILE_TO_PROCESS[$file_num]=$dfltfile_1
499:         ((++file_num))
500:     else
501:         file_notfound=true
502:     fi
503:     # Niet verplicht aanwezig.
504:     if [[ -e $dfltfile_2 ]]; then
505:         FILE_TO_PROCESS[$file_num]=$dfltfile_2
506:         ((++file_num))
507:     fi
508:     if $file_notfound; then
509:         info "Geen installatiebestanden gevonden voor $DISTRO $VERSION \
510: $EDITION."
511:         exit $WARNING
512:     fi
513: }
514:
515:
516: process_option_apps() {
517:     # Install zoekt naar apps in alle kz-install-<distro>-<versie>-bestanden
518:     # en stopt bij de eerstgevonden (installatie is computer-onafhankelijk),
519:     # setup zoekt naar apps in kz-install-<distro>-<versie>-<edition>.sh en
520:     # kz-install-<distro>-<versie>-<edition>-<computernaam>.sh.
521:     process_dfltfiles
522:     printf '%s\n' "De volgende apps zijn beschikbaar:
523:
524: NUMMER APP
525: $(
526:     if ! grep --no-messages \
527:         --regexp='^#1 ' \
528:         "${FILE_TO_PROCESS[@]}" \
529:         awk '{print $2}' \
530:         sort --unique \
531:         nl --number-width=6 \
532:         --number-format=rn \
533:         --number-separator=' ' \
534:         --body-numbering=a; then
535:         printf '%s\n' " 0 Geen apps gevonden voor $DISTRO $VERSION."
536:     else
537:         printf '%s\n' "
538: Om APPs te bekijken voer uit:
539:   ${BLUE}$DISPLAY_NAME --sim APP...${NORMAL}
540:   ${BLUE}$DISPLAY_NAME --cat APP...${NORMAL}
541: Om APPs in te stellen voer uit:
542:   ${BLUE}$DISPLAY_NAME APP...${NORMAL}"
543:     fi
544: )"

```



```

545: }
546:
547:
548: process_option_files() {
549:     printf '%s\n' "De volgende instellingsbestanden zijn beschikbaar:
550:
551: NUMMER BESTAND
552: $(
553:     if ! find "${FILE_TO_PROCESS[@]}" \
554:         2> /dev/null |
555:         nl --number-width=6 \
556:            --number-format=rn \
557:            --number-separator=' ' \
558:            --body-numbering=a; then
559:         printf '%s\n' "    0 Geen instellingsbestanden gevonden voor $DISTRO \
560: $VERSION $EDITION.
561:
562: Typ 'man $DISPLAY_NAME' voor meer informatie."
563:     else
564:         printf '%s\n' "
565: Om BESTANDen te bekijken voer uit:
566:     ${BLUE}$DISPLAY_NAME --sim BESTAND...${NORMAL}
567:     ${BLUE}$DISPLAY_NAME --cat BESTAND...${NORMAL}
568: Om BESTANDen te gebruiken voer uit:
569:     ${BLUE}$DISPLAY_NAME BESTAND...${NORMAL}"
570:     fi
571: )"
572: }
573:
574:
575: process_option_cat() {
576:     for file in "${FILE_TO_PROCESS[@]}; do
577:     {
578:         printf '%s\n' "${BLUE}# $file${NORMAL}"
579:         cat "$file"
580:         printf '\n'
581:     } >> "$CMDS_FILE"
582:     done
583:     if $OPTION_GUI; then
584:         TITLE='Inhoud instellingsbestanden'
585:         if ! zenity --text-info \
586:             --width 850 \
587:             --height 400 \
588:             --title "$TITLE" \
589:             --filename "$CMDS_FILE" \
590:             --font 'Ubuntu Mono 12' \
591:             --ok-label 'OkÃ@' \
592:             --cancel-label 'Annuleren' 2> >($LOGCMD); then
593:             true
594:         fi
595:     else
596:         less "$LESS_OPTIONS" "$CMDS_FILE"
597:     fi
598: }
599:
600:
601: process_file() {
602:     local description=''
603:     local -i cmd_seq_num=0
604:     local prev_recordtype=''
605:     local record=''
606:     local recordtype=''
607:     local write_description_line=true
608:
609:     if $EXECUTE_COMMANDS; then
610:         printf '%b' "${CURSOR_INVISABLE}"
611:     fi
612:     $skip_line && printf '\n'; skip_line=true

```

```

613:
614: while read -r record; do
615:     recordtype=${record:0:2}
616:     case $recordtype in
617:         ''|#1')
618:             # Lege regel of lege app-naam.
619:             continue
620:             ;;
621:         '#2')
622:             process_description "$record"
623:             ;;
624:         '#3')
625:             process_setup_instruction "$record"
626:             ;;
627:         '#4')
628:             process_reset_instruction "$record"
629:             ;;
630:         '#'*')
631:             # Commentaarregel.
632:             continue
633:             ;;
634:         *)
635:             process_command "$record"
636:             ;;
637:     esac
638:     prev_recordtype=$recordtype
639: done < "$CMDS_FILE"
640:
641: if $EXECUTE_COMMANDS; then
642:     printf '%b' "${CURSOR_VISABLE}"
643: fi
644: }
645:
646:
647: process_description() {
648:     local record=${1:-record?}
649:
650:     description="${record:3}"
651:     if ! $OPTION_SIMULATE && ! $EXECUTE_COMMANDS; then
652:         return $SUCCESS
653:     fi
654:
655:     write_description_line=true
656:     ((++app_seq_num))
657:     cmd_seq_num=0
658: }
659:
660:
661: process_setup_instruction() {
662:     local record=${1:-record?}
663:     local filler=''
664:     local -i len=0
665:
666:     INFO_FOUND=true
667:     if ! $OPTION_INFO; then
668:         return $SUCCESS
669:     fi
670:     filler=${SPACES:0:8}
671:     if [[ $prev_recordtype = '#3' ]]; then
672:         printf '%s%\n\n' "$filler" "${record:3}" >> "$TEXT_FILE"
673:     else
674:         ((++app_seq_num))
675:         len=${#description}
676:         # ?? wordt in term_script vervangen door aantal in te stellen apps.
677:         printf "\n[%2d/??] %s\n%s%.${len}s\n\n%s%\n\n" \
678:             "$app_seq_num" "$description" \
679:             "$filler" "$DASHES" \
680:             "$filler" "${record:3}" >> "$TEXT_FILE"

```

```

681:     fi
682: }
683:
684:
685: process_reset_instruction() {
686:     local      record=${1:-record?}
687:     local      filler=''
688:     local      -i len=0
689:
690:     RESET_FOUND=true
691:     if ! $OPTION_RESET; then
692:         return $SUCCESS
693:     fi
694:     filler=${SPACES:0:8}
695:     if [[ $prev_recordtype = '#4' ]]; then
696:         printf '%s\n\n' "$filler" "${record:3}" >> "$TEXT_FILE"
697:     else
698:         ((++app_seq_num))
699:         len=${#description}
700:         # ?? wordt in term_script vervangen door aantal te resetten apps.
701:         printf "\n[%2d/??] %s\n%s%.${len}s\n\n%s\n\n" \
702:             "$app_seq_num" "$description" \
703:             "$filler" "$DASHES" \
704:             "$filler" "${record:3}" >> "$TEXT_FILE"
705:     fi
706: }
707:
708:
709: process_command() {
710:     local      record=${1:-record?}
711:     local      -i cmd_rc=0
712:     local      description_line=''
713:     local      -i desc_maxlen=1000
714:     local      desc_printf=''
715:     local      status=''
716:
717:     COMMANDS_FOUND=true
718:     ((++cmd_seq_num))
719:
720:     if $OPTION_SIMULATE; then
721:         if $write_description_line; then
722:             printf "[%3d/%-3d] %s\n\n" \
723:                 "$app_seq_num" \
724:                 "$app_tot_num" \
725:                 "$description" >> "$TEXT_FILE"
726:             write_description_line=false
727:         fi
728:         printf '%9s [%2d] %s\n\n' \
729:             " " \
730:             "$cmd_seq_num" \
731:             "${BLUE}$record${NORMAL}" >> "$TEXT_FILE"
732:     elif $EXECUTE_COMMANDS; then
733:         desc_printf=$description
734:         if $OPTION_GUI; then
735:             TEXT="#[$app_seq_num/$app_tot_num] $description...\n\n$record"
736:             printf '%s\n' "$TEXT"
737:         else
738:             if [[ -t 1 ]]; then
739:                 # Afbreken in Terminalvenster moet i.v.m. updaten status.
740:                 # <----- tput cols ----->
741:                 # '[WERK] [ 1/1 ] [ 1] Te lange descr...' <=== op 1 regel
742:                 # <----- 22 ----->
743:                 desc_maxlen=$((tput cols) - 22)
744:                 if [[ ${#desc_printf} -gt $desc_maxlen ]]; then
745:                     desc_printf=${desc_printf:0:((desc_maxlen - 3))}'...'
746:                 fi
747:             fi
748:         fi

```

```

749: printf -v description_line \
750:      "[%3d/%-3d] [%2d] %-${desc_maxlen}.${desc_maxlen}s" \
751:      "$app_seq_num" \
752:      "$app_tot_num" \
753:      "$cmd_seq_num" \
754:      "$desc_printf"
755: execute_command "$record"
756: fi
757: }
758:
759:
760: execute_command() {
761:     local cmd=${1:-cmd?}
762:     local -i cmd_rc=0
763:     local filler=''
764:     local status=''
765:
766:     if ! $OPTION_GUI; then
767:         if [[ $cmd_seq_num -eq 1 ]]; then
768:             printf "$STATUS_BUSY %s${CARRIAGE_RETURN}" "$description_line"
769:         else
770:             printf "$STATUS_BUSY ${SPACES:0:9}%s${CARRIAGE_RETURN}" \
771:                 "${description_line:9}"
772:         fi
773:     fi
774:
775:     log "$description_line"
776:     log "${BLUE}$cmd${NORMAL}"
777:     eval "$cmd" |& $LOGCMD || cmd_rc=$?
778:     if [[ $cmd_rc -gt $MAXRC ]]; then
779:         MAXRC=$cmd_rc
780:     fi
781:     if [[ $cmd_rc -eq $SUCCESS ]]; then
782:         status="$STATUS_SUCCESS"
783:     else
784:         status="$STATUS_ERROR"
785:     fi
786:     log "$status rc=$cmd_rc, maxrc=$MAXRC"
787:     printf '%s\n' "$status"
788: }
789:
790:
791: term_script() {
792:     local -i app_num=0
793:
794:     if $OPTION_SIMULATE; then
795:         if $OPTION_GUI; then
796:             TITLE='Instellingen-opdrachten'
797:             if ! zenity --text-info \
798:                 --width 1000 \
799:                 --height 400 \
800:                 --title "$TITLE" \
801:                 --filename "$TEXT_FILE" \
802:                 --font 'Ubuntu Mono 12' \
803:                 --ok-label 'OkÃ©' \
804:                 --cancel-label 'Annuleren' 2> >($LOGCMD); then
805:                 true
806:             fi
807:         else
808:             less "$LESS_OPTIONS" "$TEXT_FILE"
809:         fi
810:     elif $OPTION_INFO && $INFO_FOUND; then
811:         printf '\n' >> "$TEXT_FILE"
812:         # Vervang ?? door aantal in te stellen apps.
813:         app_num=$(grep --regexp='\??' ' --count "$TEXT_FILE")
814:         sed -i "s|/??| |/${printf %-2d "$app_num"}" |g" "$TEXT_FILE"
815:         if $OPTION_GUI; then
816:             TITLE='Instel-instructies'
```

```

817:         if ! zenity --text-info \
818:             --width          850 \
819:             --height         400 \
820:             --title          "$TITLE" \
821:             --filename       "$TEXT_FILE" \
822:             --font            'Ubuntu Mono 12' \
823:             --ok-label        'OkÃ©' \
824:             --cancel-label    'Annuleren'      2> >($LOGCMD); then
825:             true
826:         fi
827:     else
828:         less "$LESS_OPTIONS" "$TEXT_FILE"
829:     fi
830: elif $OPTION_INFO && ! $INFO_FOUND; then
831:     info 'Geen instel-instructies.'
832: elif $OPTION_RESET && $RESET_FOUND; then
833:     printf '\n' >> "$TEXT_FILE"
834:     # Vervang ?? door aantal te resetten apps.
835:     app_num=$(grep --regexp='\??' ' --count "$TEXT_FILE")
836:     sed -i "s|/??| |/${printf %-2d "$app_num"}] |g" "$TEXT_FILE"
837:     # Vervang 'instellen' door 'resetten' (1 letter korter).
838:     filler=${SPACES:0:8}
839:     sed --in-place --expression='s/instellen/resetten/g' "$TEXT_FILE"
840:     sed --in-place --expression="s/^$filler==/$filler=/g" "$TEXT_FILE"
841:     if $OPTION_GUI; then
842:         TITLE='Reset-instructies'
843:         if ! zenity --text-info \
844:             --width          850 \
845:             --height         400 \
846:             --title          "$TITLE" \
847:             --filename       "$TEXT_FILE" \
848:             --font            'Ubuntu Mono 12' \
849:             --ok-label        'OkÃ©' \
850:             --cancel-label    'Annuleren'      2> >($LOGCMD); then
851:             true
852:         fi
853:     else
854:         less "$LESS_OPTIONS" "$TEXT_FILE"
855:     fi
856: elif $OPTION_RESET && ! $RESET_FOUND; then
857:     info 'Geen reset-instructies.'
858: elif ! $COMMANDS_FOUND; then
859:     info 'Geen opdrachten.'
860: fi
861:
862: if $EXECUTE_COMMANDS && $INFO_FOUND && ! $OPTION_GUI; then
863:     info "
864: Er zijn aanvullende instructies!
865: Gebruik de opdracht:
866: ${BLUE}$DISPLAY_NAME --info ${APP_ARGUMENT[*]}${NORMAL}"
867: fi
868:
869: if $EXECUTE_COMMANDS && $OPTION_GUI; then
870:     if [[ $MAXRC -ne $SUCCESS ]]; then
871:         warning 'Instellen is geÃ«indigd met een waarschuwing.'
872:         exit $WARNING
873:     else
874:         info 'Instellen is voltooid.'
875:         exit $SUCCESS
876:     fi
877: fi
878: }
879:
880:
881: #####
882: # Main line
883: #####
884:

```

```
885: main() {
886:     init_script "$@"
887:     check_input "$@"
888:     process_input
889:     term_script
890: }
891:
892:
893: main "$@"
894:
895:
896: # EOF
```