```
#!/bin/bash
# shellcheck source=kzcommon.sh
# ------------------------------------------------------------------------------
# Apps instellen.
#
# Geschreven door Karel Zimmer <info@karelzimmer.nl>.
#
# Auteursrecht (c) 2013-2021 Karel Zimmer.
# GNU Algemene Publieke Licentie <https://www.gnu.org/licenses/gpl.html>.
#
# RelNum=41.11.01
# RelDat=2021-01-19
# ------------------------------------------------------------------------------

# ------------------------------------------------------------------------------
# Global constants
# ------------------------------------------------------------------------------
source "$(dirname "$0")"/kzcommon.sh
readonly BEGIN_INFO="\
$DASHES
-  Volg onderstaande aanvullende instructies na het instellen van deze apps   -
$DASHES"
readonly END_INFO="
$DASHES
-  Volg bovenstaande aanvullende instructies na het instellen van deze apps   -
$DASHES"
readonly BEGIN_RESET_INFO="\
$DASHES
-       Volg onderstaande instructies voor het herstellen van deze apps       -
$DASHES"
readonly END_RESET_INFO="
$DASHES
-       Volg bovenstaande instructies voor het herstellen van deze apps       -
$DASHES"
readonly EDITION_DEFAULT=desktop

# Bij aanpassingen ook .completion aanpassen!
readonly OPTIONS_SHORT=$OPTIONS_SHORT_COMMON'ace:firs'
readonly OPTIONS_LONG=$OPTIONS_LONG_COMMON",apps,cat,edition:,files,info,
restore-info,simulate"
readonly USAGE="Gebruik: $PROGNAME [-a|--apps] [-c|--cat] \
[-e|--edition=EDITIE] [-f|--files]
                 [-i|--info] [-r|--restore-info] [-s|--simulate]
                 $OPTIONS_USAGE_COMMON
                 [APP...] [BESTAND...]"
readonly HELP="Gebruik: $PROGNAME [OPTIE...] [--] [APP...] [BESTAND...]

Apps instellen.

Opties:
  -a --apps     toon lijst van in te stellen apps
  -c --cat      toon inhoud van instellingsbestanden
  -e --edition=EDITIE
                gebruik opgegeven editie
  -f --files    toon lijst van alle instellingsbestanden
  -i --info     toon aanvullende instructies na het instellen van de apps
  -r --restore-info
                toon instructies voor het herstellen van de apps
  -s --simulate
                geen actie, simuleer wijzigingen, en toon opdracht die
                uitgevoerd zou worden
$OPTIONS_HELP_COMMON

Argumenten:
  APP           opgegeven apps verwerken
  BESTAND       opgegeven bestanden verwerken"

readonly STATUS_BUSY="[${BLINK}${BOLD}WERK${NORMAL}]"
```

```
readonly STATUS_ERROR="[${RED}FOUT${NORMAL}]"
readonly STATUS_SUCCESS="[${GREEN}GOED${NORMAL}]"

# Terminalattributen ('man terminfo'). Gebruik ${<variabele_naam>}.
readonly BLINK=$(tput blink)
readonly CURSOR_INVISABLE=$(tput civis)

# ------------------------------------------------------------------------------
# Global variables
# ------------------------------------------------------------------------------
declare -a APP_ARGUMENT=''
declare -a FILE_ARGUMENT=''
declare -a INPUTFILE=''
declare ARGUMENT_APP=false
declare ARGUMENT_FILE=false
declare COMMANDS_FOUND=false
declare DISTRO=''
declare DFLTFILE_1=''
declare DFLTFILE_2=''
declare DFLTFILE_NOTFOUND=''
declare EDITION_ARGUMENT=''
declare EDITION=$EDITION_DEFAULT
declare EXECUTE_COMMANDS=true
declare -i MAXRC=0
declare INFO_FOUND=false
declare OPTION_APPS=false
declare OPTION_CAT=false
declare OPTION_EDITION=false
declare OPTION_FILES=false
declare OPTION_INFO=false
declare OPTION_RESTORE_INFO=false
declare OPTION_SIMULATE=false
declare RESET_INFO_FOUND=false
declare TEMP_TEXT_FILE=''

# ------------------------------------------------------------------------------
# Functions
# ------------------------------------------------------------------------------
check_input() {
    local file_notfound=false
    local -i app_arg_num=0
    local -i file_arg_num=0
    local -i file_num=0
    local -i getopt_rc=0
    local parsed=''

    parsed=$(
        getopt  --alternative                       \
                --options       "$OPTIONS_SHORT"    \
                --longoptions   "$OPTIONS_LONG"     \
                --name          "$PROGNAME"         \
                -- "$@"
        )       || getopt_rc=$?
    if [[ $getopt_rc -ne 0 ]]; then
        printf '%s\n' "$USAGELINE" >&2
        exit $ERROR
    fi
    eval set -- "$parsed"
    process_general_options "$@"

    while true; do
        case $1 in
            -a|--apps)
                OPTION_APPS=true
                shift
                ;;
            -c|--cat)
                OPTION_CAT=true
```

```
137:                 shift
138:                 ;;
139:             -e|--edition)
140:                 if $OPTION_EDITION; then
141:                     printf "$PROGNAME: %s\n%s\n" "optie ’$1’ Ã©Ã©nmaal opgeven" \
142: "$USAGELINE" >&2
143:                     exit $ERROR
144:                 else
145:                     OPTION_EDITION=true
146:                     EDITION_ARGUMENT=$2
147:                     EDITION=$EDITION_ARGUMENT
148:                 fi
149:                 shift 2
150:                 ;;
151:             -f|--files)
152:                 OPTION_FILES=true
153:                 shift
154:                 ;;
155:             -g|--gui)
156:                 OPTION_GUI=true
157:                 shift
158:                 ;;
159:             -i|--info)
160:                 OPTION_INFO=true
161:                 EXECUTE_COMMANDS=false
162:                 shift
163:                 ;;
164:             -r|--restore-info)
165:                 OPTION_RESTORE_INFO=true
166:                 EXECUTE_COMMANDS=false
167:                 shift
168:                 ;;
169:             -s|--simulate)
170:                 OPTION_SIMULATE=true
171:                 EXECUTE_COMMANDS=false
172:                 shift
173:                 ;;
174:             --)
175:                 shift
176:                 break
177:                 ;;
178:             *)
179:                 shift
180:                 ;;
181:         esac
182:     done
183:
184:     id=$(lsb_release --id --short | tr ’[:upper:]’ ’[:lower:]’)
185:     release=$(lsb_release --release --short)
186:     DISTRO=$id-$release-$EDITION
187:     DFLTFILE_1=$PROGDIR/$PROGNAME-$DISTRO.sh
188:     DFLTFILE_2=$PROGDIR/$PROGNAME-$DISTRO-$HOSTNAME.sh
189:     DFLTFILE_NOTFOUND="Geen bestanden zoals $PROGNAME-$DISTRO*.sh in $PROGDIR."
190:
191:     # Verwerk argumenten.
192:     while [[ "$*" ]]; do
193:         if [[ "$(basename "$1")" = $PROGNAME-*.sh ]]; then
194:             ARGUMENT_FILE=true
195:             FILE_ARGUMENT[$file_arg_num]=$1
196:             ((++file_arg_num))
197:             shift
198:         else
199:             ARGUMENT_APP=true
200:             APP_ARGUMENT[$app_arg_num]=$1
201:             ((++app_arg_num))
202:             shift
203:         fi
204:     done
```

```
  205:     if $ARGUMENT_APP; then
  206:         process_argument_app
  207:     fi
  208:     if $ARGUMENT_FILE; then
  209:         process_argument_file
  210:     fi
  211:     # Als APP of FILE is opgegeven, de DFLTFILEs niet verwerken.
  212:     if ! ($ARGUMENT_APP || $ARGUMENT_FILE); then
  213:         process_dfltfiles
  214:     fi
  215:     if $file_notfound; then
  216:         exit $WARNING
  217:     fi
  218:
  219:     # Verwerk opties.
  220:     if $OPTION_APPS; then
  221:         process_option_apps
  222:         exit $SUCCESS
  223:     elif $OPTION_FILES; then
  224:         process_option_files
  225:         exit $SUCCESS
  226:     # Optie cat is afhankelijk van argumenten APP en FILE.
  227:     elif $OPTION_CAT; then
  228:         process_option_cat
  229:         exit $SUCCESS
  230:     fi
  231:
  232:     # We gaan opdrachten uitvoeren, mogen we dat wel?
  233:     if $EXECUTE_COMMANDS; then
  234:         check_user
  235:     fi
  236: }
  237:
  238: process_option_apps() {
  239:     # shellcheck disable=SC2062
  240:     printf '%s\n' "De volgende apps kunnen ingesteld worden:
  241: NUMMER APP
  242: $(
  243:     if !    grep    --regexp='^#@'                      \
  244:                     --no-messages                       \
  245:                     "$PROGDIR/$PROGNAME-$DISTRO"*.sh    |
  246:             awk     -F#@                                \
  247:                     '{print $2}'                        |
  248:             sort    --unique                            |
  249:             nl      --number-width=6                    \
  250:                     --number-format=rn                  \
  251:                     --number-separator=' '              \
  252:                     --body-numbering=a; then
  253:         printf '%s\n' "     0 Geen apps gevonden.
  254:
  255: $DFLTFILE_NOTFOUND"
  256:     else
  257:         printf '%s\n' "
  258: De gevonden apps zijn te gebruiken als invoer voor dit script:
  259:     ${BLUE}$PROGNAME APP...${NORMAL}
  260: Voor meer informatie voer uit:
  261:     ${BLUE}$PROGNAME --sim APP...${NORMAL}
  262:     ${BLUE}$PROGNAME --cat APP...${NORMAL}"
  263:     fi
  264: )"
  265: }
  266:
  267: process_option_files() {
  268:     # shellcheck disable=SC2012
  269:     printf '%s\n' "De volgende instellingsbestanden zijn beschikbaar:
  270: NUMMER BESTAND
  271: $(
  272:     if !    ls  --format=single-column                  \
```

```
                "$PROGDIR/$PROGNAME-$DISTRO"*.sh            \
                2> /dev/null                                |
            nl  --number-width=6                            \
                --number-format=rn                          \
                --number-separator=' '                      \
                --body-numbering=a; then
        printf '%s\n' "     0 Geen bestanden gevonden.

$DFLTFILE_NOTFOUND"
    else
        printf '%s\n' "
De gevonden bestanden zijn te gebruiken als invoer voor dit script:
    ${BLUE}$PROGNAME BESTAND...${NORMAL}
Voor meer informatie voer uit:
    ${BLUE}$PROGNAME --sim BESTAND...${NORMAL}
    ${BLUE}$PROGNAME --cat BESTAND...${NORMAL}"
    fi
)"
}

process_argument_app() {
    local app=''
    local file=''

    for app in "${APP_ARGUMENT[@]}"; do
        # Zoek exact (line-regexp) naar app-tag in instellingsbestanden.
        # Alleen eerstgevonden bestand verwerken (lines=1), want tag kan in
        # meerdere bestanden voorkomen.
        # shellcheck disable=SC2062
        file=$(
            if ! grep   --files-with-matches                \
                        --line-regexp                       \
                        --regexp="#@$app"                   \
                        "$PROGDIR/$PROGNAME-$DISTRO"*.sh    \
                        2> /dev/null                        |
                head    --lines=1; then
                printf '%s' ''
            fi
            )
        if [[ $file ]]; then
            process_argument_app_file
        else
            info "App '$app' niet gevonden."
            file_notfound=true
        fi
    done
    if $file_notfound; then
        info "Typ 'kzsetup --apps' voor meer informatie."
    fi
}

process_argument_app_file() {
    local app_found=false
    local description_found=false
    local temp_sh=''
    local record=''

    temp_sh=$(mktemp -t "$PROGNAME-$app-XXXXXXXXXX.sh")
    while read -r record; do
        case $record in
            '#@'*)
                if [[ $record = '#@'$app ]]; then
                    if $app_found; then
                        info "Dubbele app-tag $app."
                    else
                        # Gezochte app-tag.
                        printf '%b\n' "# $DASHES\n# Bestand: $temp_sh" >> \
"$temp_sh"
```

```
                        # Regels doorschrijven naar tijdelijk bestand tot
                        # eerstvolgende app-tag of beschrijving (verplicht), of
                        # tot EOF.
                        app_found=true
                    fi
                elif $app_found; then
                    # Volgende app-tag.
                    break
                fi
                ;;
            '#1'*)
                # Beschrijving.
                if $description_found; then
                    # Volgende beschrijving.
                    break
                elif $app_found; then
                    printf '%b\n' "#    Bron: $file\n# $DASHES\n\n$record" >> \
"$temp_sh"
                    description_found=true
                fi
                ;;
            *)
                # Overige regels.
                if $app_found; then
                    printf '%s\n' "$record" >> "$temp_sh"
                fi
                ;;
        esac
    done < "$file"
    if ! grep   --quiet         \
                --line-regexp   \
                --regexp='^# EOF' "$temp_sh"
    then
        printf '%s\n' '# EOF' >> "$temp_sh"
    fi
    INPUTFILE[$file_num]=$temp_sh
    ((++file_num))
}

process_argument_file() {
    local file=''

    for file in "${FILE_ARGUMENT[@]}"; do
        if [[ -e $file ]]; then
            INPUTFILE[$file_num]=$file
            ((++file_num))
        else
            info "Bestand '$file' niet gevonden."
            file_notfound=true
        fi
    done
    if $file_notfound; then
        info "Typ 'kzsetup --files' voor meer informatie."
    fi
}

process_dfltfiles() {
    if [[ -e $DFLTFILE_1 ]]; then
        INPUTFILE[$file_num]=$DFLTFILE_1
        ((++file_num))
    else
        file_notfound=true
    fi
    # Niet verplicht aanwezig.
    if [[ -e $DFLTFILE_2 ]]; then
        INPUTFILE[$file_num]=$DFLTFILE_2
        ((++file_num))
    fi
```

```
409:     if $file_notfound; then
410:         info "$DFLTFILE_NOTFOUND"
411:     fi
412: }
413:
414: process_option_cat() {
415:     local skip_line=false
416:     local temp_text_file
417:
418:     temp_text_file=$(mktemp -t "$PROGNAME-XXXXXXXXXX.txt")
419:
420:     for file in "${INPUTFILE[@]}"; do
421:         {
422:             $skip_line && printf '\n'; skip_line=true
423:             printf '%s\n' "${BLUE}# $file${NORMAL}"
424:             cat "$file"
425:         }   >> "$temp_text_file"
426:     done
427:     less "$LESS_OPTIONS" "$temp_text_file"
428:     rm "$temp_text_file"
429: }
430:
431: process_input() {
432:     local file=''
433:     local -i app_seq_num=0
434:     local -i app_tot_num=0
435:     local skip_line=false
436:     local text=''
437:     local title=''
438:
439:     TEMP_TEXT_FILE=$(mktemp -t "$PROGNAME-XXXXXXXXXX.txt")
440:
441:     for file in "${INPUTFILE[@]}"; do
442:         if $OPTION_SIMULATE || $EXECUTE_COMMANDS; then
443:             app_seq_num=0
444:             # 'if ! grep --regexp='^#1' ...; then ...' werkt niet; grep geeft
445:             # 0 terug als niets gevonden, OK, maar dan ook rc=1, Â¬OK.
446:             app_tot_num=$(
447:                 grep    --word-regexp   \
448:                         --regexp='^#1'  \
449:                         --count "$file" ||
450:                 true
451:                 )
452:         fi
453:
454:         if $EXECUTE_COMMANDS && $OPTION_GUI; then
455:             title='Apps instellen'
456:             text="Verwerk bestand $file"
457:             # Met "|& zenity --progress" worden globale variabelen uit
458:             # aangeroepen functies niet doorgegeven, vandaar de
459:             # 'process substitution' met "> >(zenity ...)".
460:             # Zenity progress-informatie per bestand (kzinstall per opdracht).
461:             process_file "$file" > >(
462:                 zenity  --progress              \
463:                         --pulsate               \
464:                         --auto-close            \
465:                         --no-cancel             \
466:                         --width     700         \
467:                         --height    50          \
468:                         --title     "$title"    \
469:                         --text      "$text"     \
470:                         2> >($LOGCMD)
471:                 )
472:         else
473:             process_file "$file"
474:         fi
475:     done
476: }
```

```
477:
478: process_file() {
479:     local file=${1:-file?}
480:     local description='### GEEN BESCHRIJVING ###'
481:     local file_text=''
482:     local -i cmd_seq_num=0
483:     local prev_recordtype=''
484:     local record=''
485:     local recordtype=''
486:     local write_description_line=true
487:
488:     file_text="${BOLD}[BEGIN] Bestand $file${NORMAL}"
489:     if $EXECUTE_COMMANDS; then
490:         log "$file_text"
491:         printf '%b'   "${CURSOR_INVISABLE}"
492:         $skip_line && printf '\n'; skip_line=true
493:         printf '%s\n' "$file_text"
494:     elif $OPTION_SIMULATE; then
495:         $skip_line && printf '\n'   >> "$TEMP_TEXT_FILE"; skip_line=true
496:         printf '%s\n' "$file_text"  >> "$TEMP_TEXT_FILE"
497:     fi
498:
499:     while read -r record; do
500:         recordtype=${record:0:2}
501:         case $recordtype in
502:             '#1')
503:                 process_description "$record"
504:                 ;;
505:             '#2')
506:                 process_setup_instruction "$record"
507:                 ;;
508:             '#3')
509:                 process_restore_instruction "$record"
510:                 ;;
511:             ''|'#'*)
512:                 continue
513:                 ;;
514:             *)
515:                 process_command "$record"
516:                 ;;
517:         esac
518:         prev_recordtype=$recordtype
519:     done < "$file"
520:
521:     file_text="${BOLD}[EINDE] Bestand $file${NORMAL}"
522:     if $EXECUTE_COMMANDS; then
523:         log "$file_text"
524:         printf '%s\n' "$file_text"
525:         printf '%b'   "${CURSOR_VISABLE}"
526:     elif $OPTION_SIMULATE; then
527:         printf '%s\n' "$file_text" >> "$TEMP_TEXT_FILE"
528:     fi
529: }
530:
531: process_description() {
532:     local record=${1:-record?}
533:
534:     description="${record:3}"
535:     if ! $OPTION_SIMULATE && ! $EXECUTE_COMMANDS; then
536:         return 0
537:     fi
538:
539:     write_description_line=true
540:     ((++app_seq_num))
541:     cmd_seq_num=0
542: }
543:
544: process_setup_instruction() {
```

```
545:     local record=${1:-record?}
546:
547:     INFO_FOUND=true
548:     if ! $OPTION_INFO; then
549:         return 0
550:     fi
551:     if [[ $app_seq_num -eq 0 ]]; then
552:         printf '%s\n' "$BEGIN_INFO" > "$TEMP_TEXT_FILE"
553:     fi
554:     if [[ $prev_recordtype = '#2' ]]; then
555:         printf '%s\n' "    ${record:3}" >> "$TEMP_TEXT_FILE"
556:     else
557:         ((++app_seq_num))
558:         printf "\n%2s. %s\n    %.${#description}s\n    %s\n" "$app_seq_num" \
559: "$description" "$DASHES" "${record:3}" >> "$TEMP_TEXT_FILE"
560: #        printf "\n%2s. %s\n    %s\n" "$app_seq_num" "$description" \
561: #"${record:3}" >> "$TEMP_TEXT_FILE"
562:     fi
563: }
564:
565: process_restore_instruction() {
566:     local record=${1:-record?}
567:
568:     RESET_INFO_FOUND=true
569:     if ! $OPTION_RESTORE_INFO; then
570:         return 0
571:     fi
572:     if [[ $app_seq_num -eq 0 ]]; then
573:         printf '%s\n' "$BEGIN_RESET_INFO" > "$TEMP_TEXT_FILE"
574:     fi
575:     if [[ $prev_recordtype = '#3' ]]; then
576:         printf '%s\n' "    ${record:3}" >> "$TEMP_TEXT_FILE"
577:     else
578:         ((++app_seq_num))
579:         printf "\n%2s. %s\n    %.${#description}s\n    %s\n" "$app_seq_num" \
580: "$description" "$DASHES" "${record:3}" >> "$TEMP_TEXT_FILE"
581: #        printf "\n%2s. %s\n    %.${#description}s\n    %s\n" "$app_seq_num" \
582: #"$description" "${record:3}" >> "$TEMP_TEXT_FILE"
583:     fi
584: }
585:
586: process_command() {
587:     local record=${1:-record?}
588:     local -i cmd_rc=0
589:     local description_line=''
590:     local -i description_maxlen=1000
591:     local description_printf=''
592:     local status=''
593:
594:     COMMANDS_FOUND=true
595:     ((++cmd_seq_num))
596:
597:     if $OPTION_SIMULATE; then
598:         if $write_description_line; then
599:             printf '[%2d/%-2d] %s\n' \
600: "$app_seq_num" "$app_tot_num" "$description" >> "$TEMP_TEXT_FILE"
601:         fi
602:         write_description_line=false
603:         printf '%7s [%2d] %s\n' " " "$cmd_seq_num" "${BLUE}$record${NORMAL}" \
604: >> "$TEMP_TEXT_FILE"
605:     elif $EXECUTE_COMMANDS; then
606:         description_printf=$description
607:         if ! $OPTION_GUI; then
608:             # Afbreken in Terminalvenster moet i.v.m. updaten status van BEZIG
609:             # naar GOED of FOUT.
610:             #  <------------ tput cols ------------->
611:             # '[BEZIG] [ 1/1 ] [ 1] Te lange descr...' <=== op 1 regel blijven
612:             #  <------- 21 -------->
```

```
  613:             description_maxlen=$(($(tput cols) - 21))
  614:             description_printf=$description
  615:             if [[ ${#description_printf} -gt $description_maxlen ]]; then
  616:                 description_printf=\
  617: ${description_printf:0:(($description_maxlen - 3))}'...'
  618:             fi
  619:         fi
  620:         printf -v description_line \
  621: "[%2d/%-2d] [%2d] %-${description_maxlen}.${description_maxlen}s" \
  622: "$app_seq_num" "$app_tot_num" "$cmd_seq_num" "$description_printf"
  623:         execute_command "$record"
  624:     fi
  625: }
  626:
  627: execute_command() {
  628:     local cmd=${1:-cmd?}
  629:     local -i cmd_rc=0
  630:     local status=''
  631:
  632:     if ! $OPTION_GUI; then
  633:         if [[ $cmd_seq_num -eq 1 ]]; then
  634:             printf " $STATUS_BUSY %s\r" "$description_line"
  635:         else
  636:             printf " $STATUS_BUSY         %s\r" "${description_line:8}"
  637:         fi
  638:     fi
  639:
  640:     log "$description_line"
  641:     log "${BLUE}$cmd${NORMAL}"
  642:     eval "$cmd" |& $LOGCMD || cmd_rc=$?
  643:     if [[ $cmd_rc -gt $MAXRC ]]; then
  644:         MAXRC=$cmd_rc
  645:     fi
  646:     if [[ $cmd_rc -eq 0 ]]; then
  647:         status="$STATUS_SUCCESS"
  648:     else
  649:         status="$STATUS_ERROR"
  650:     fi
  651:
  652:     printf ' %s\n' "$status"
  653:     log "$status rc=$cmd_rc, maxrc=$MAXRC"
  654: }
  655:
  656: term_script() {
  657:     local title=''
  658:
  659:     if $OPTION_SIMULATE; then
  660:         cat "$TEMP_TEXT_FILE"
  661:     elif $OPTION_INFO && $INFO_FOUND; then
  662:         printf '%s\n' "$END_INFO" >> "$TEMP_TEXT_FILE"
  663:         if $OPTION_GUI; then
  664:             title='Instel-instructies'
  665:             if ! zenity --text-info                          \
  666:                         --width         850                  \
  667:                         --height        400                  \
  668:                         --title         "$title"             \
  669:                         --filename      "$TEMP_TEXT_FILE"    \
  670:                         --font          'Ubuntu Mono 12'     \
  671:                         --ok-label      'OkÃ©'                \
  672:                         --cancel-label  'Annuleren'          \
  673:                         2> >($LOGCMD); then
  674:                 true
  675:             fi
  676:         else
  677:             less "$LESS_OPTIONS" "$TEMP_TEXT_FILE"
  678:         fi
  679:     elif $OPTION_INFO && ! $INFO_FOUND; then
  680:         info 'Geen instructies aanwezig.'
```

```
681:     elif $OPTION_RESTORE_INFO && $RESET_INFO_FOUND; then
682:         printf '%s\n' "$END_RESET_INFO" >> "$TEMP_TEXT_FILE"
683:         if $OPTION_GUI; then
684:             title='Herstel-instructies'
685:             if ! zenity --text-info                         \
686:                         --width         850                 \
687:                         --height        400                 \
688:                         --title         "$title"            \
689:                         --filename      "$TEMP_TEXT_FILE"   \
690:                         --font          'Ubuntu Mono 12'    \
691:                         --ok-label      'OkÃ©'               \
692:                         --cancel-label  'Annuleren'         \
693:                         2> >($LOGCMD); then
694:                 true
695:             fi
696:         else
697:             less "$LESS_OPTIONS" "$TEMP_TEXT_FILE"
698:         fi
699:     elif $OPTION_RESTORE_INFO && ! $RESET_INFO_FOUND; then
700:         info 'Geen instructies aanwezig.'
701:     elif ! $COMMANDS_FOUND; then
702:         info 'Geen opdrachten aanwezig.'
703:     fi
704:     rm "$TEMP_TEXT_FILE"
705:
706:     if  $EXECUTE_COMMANDS   &&
707:         [[ $MAXRC -eq 0 ]]  &&
708:         $INFO_FOUND         &&
709:         ! $OPTION_GUI; then
710:         info "Er zijn aanvullende instructies, gebruik de opdracht:
711:     ${BLUE}$PROGNAME --info${NORMAL}"
712:     fi
713:
714:     if $EXECUTE_COMMANDS && [[ $MAXRC -ne 0 ]]; then
715:         exit $WARNING
716:     else
717:         exit $SUCCESS
718:     fi
719: }
720:
721: # ------------------------------------------------------------------------------
722: # Main line
723: # ------------------------------------------------------------------------------
724: main() {
725:     init_script
726:     check_input "$@"
727:     process_input
728:     term_script
729: }
730:
731: main "$@"
732:
733: # EOF
```