

```
1: #!/bin/bash
2: # shellcheck source=kzcommon.sh
3: # -----
4: # Controlebestanden opsplitsen.
5: #
6: # Geschreven door Karel Zimmer <info@karelzimmer.nl>.
7: #
8: # Auteursrecht (c) 2016-2021 Karel Zimmer.
9: # GNU Algemene Publieke Licentie <https://www.gnu.org/licenses/gpl.html>.
10: #
11: # RelNum=08.01.00
12: # RelDat=2021-01-18
13: # -----
14: #
15: # -----
16: # Global constants
17: # -----
18: source "$(dirname "$0")"/kzcommon.sh
19: readonly SEARCHDIR=$HOME/Downloads
20: readonly HASHPROG=sha256sum
21:
22: # Bij aanpassingen ook .completion aanpassen!
23: readonly OPTIONS_SHORT=$OPTIONS_SHORT_COMMON
24: readonly OPTIONS_LONG=$OPTIONS_LONG_COMMON
25: readonly USAGE="Gebruik: $PROGRAMNAME $OPTIONS_USAGE_COMMON"
26: readonly HELP="Gebruik: $PROGRAMNAME [OPTIE...]"
27:
28: Controlebestanden opsplitsen.
29:
30: Opties:
31: $OPTIONS_HELP_COMMON"
32:
33: # -----
34: # Global variables
35: # -----
36: declare ISO_FILE=''
37: declare ISO_FILE_FOUND=false
38:
39: # -----
40: # Functions
41: # -----
42: check_input() {
43:     local -i getopt_rc=0
44:     local parsed=''
45:
46:     parsed=$(
47:         getopt --alternative          \
48:                --options              "$OPTIONS_SHORT"      \
49:                --longoptions         "$OPTIONS_LONG"       \
50:                --name                 "$PROGRAMNAME"       \
51:                -- "$@"
52:     ) || getopt_rc=$?
53:     if [[ $getopt_rc -ne 0 ]]; then
54:         printf '%s\n' "$USAGELINE" >&2
55:         exit $ERROR
56:     fi
57:     eval set -- "$parsed"
58:     process_general_options "$@"
59:
60:     while true; do
61:         case $1 in
62:             --)
63:                 shift
64:                 break
65:             ;;
66:             *)
67:                 shift
68:                 ;;
```

```
69:     esac
70:     done
71:
72:     if [[ "$*" ]]; then
73:         printf "$PROGNAME: %s\n%s\n" 'geen argumenten opgeven' "$USAGELINE" >&2
74:         exit $ERROR
75:     fi
76:
77:     # Een non-gui script gestart met optie gui.
78:     if $OPTION_GUI; then
79:         OPTION_GUI=false
80:         TERMINAL=true
81:     fi
82:
83:     check_user
84: }
85:
86: process_input() {
87:     cd "$SEARCHDIR"
88:
89:     for ISO_FILE in *.iso; do
90:
91:         if ! [[ -f "$ISO_FILE" ]]; then
92:             continue
93:         fi
94:
95:         ISO_FILE_FOUND=true
96:
97:         create_checksum_file
98:
99:     done
100:
101:     if ! $ISO_FILE_FOUND; then
102:         warning "Geen beeldbestanden (.iso) gevonden in '$SEARCHDIR'."
103:         rm --force -- *SHA256SUMS
104:     fi
105: }
106:
107: create_checksum_file() {
108:     local -i grep_rc=0
109:
110:     if [[ -e $ISO_FILE.sha256sum ]]; then
111:         info "Beeldbestand '$ISO_FILE' heeft al controlebestand \
112: '$ISO_FILE.sha256sum'.\n"
113:         return 0
114:     else
115:         # shellcheck disable=SC2062
116:         grep --word-regexp \
117:             --regexp="$ ISO_FILE" \
118:             --regexp="*$ISO_FILE" \
119:             --exclude=*.iso \
120:             --exclude-dir=* \
121:             --no-filename \
122:             -- * \
123:             > "$ISO_FILE".sha256sum \
124:             2> >($LOGCMD) || grep_rc=$?
125:         if [[ $grep_rc -ne 0 ]]; then
126:             error "Geen controlesom gevonden voor '$ISO_FILE'.\n"
127:             rm --force "$ISO_FILE".sha256sum
128:         else
129:             info "Controlebestand '$ISO_FILE.sha256sum' aangemaakt.\n"
130:         fi
131:     fi
132: }
133:
134: term_script() {
135:     if $ISO_FILE_FOUND; then
136:         info "Om de beeldbestanden (.iso) te controleren voer uit:
```

```
137:     ${BLUE}cd $SEARCHDIR;$HASHPROG --check *.$HASHPROG; cd -${NORMAL}
138:
139: En om vervolgens USB-sticks te maken voer uit:
140:     ${BLUE}kzmkusbs${NORMAL}
141:
142: Om de controlebestanden (.sha256sums) opnieuw te downloaden voer uit:
143:     ${BLUE}kzdlsio --checksums${NORMAL}"
144:         rm --force -- *SUM*
145:     fi
146:     exit $SUCCESS
147: }
148:
149: # -----
150: # Main line
151: # -----
152: main() {
153:     init_script
154:     check_input "$@"
155:     process_input
156:     term_script
157: }
158:
159: main "$@"
160:
161: # EOF
```