

```
1: #!/bin/bash
2: # shellcheck source=kz-common.sh
3: #####
4: # Controlebestanden opsplitsen.
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-spsum
11: DISPLAY_NAME=${PROGRAM_NAME/kz-/kz }
12: RELEASE_YEAR=2016
13:
14: VERSION_NUMBER=09.00.11
15: VERSION_DATE=2021-10-15
16:
17:
18: #####
19: # Global constants
20: #####
21:
22: readonly RUN_AS_SUPERUSER=false
23: readonly OPTIONS_SHORT=$OPTIONS_SHORT_COMMON
24: readonly OPTIONS_LONG=$OPTIONS_LONG_COMMON
25: readonly USAGE="Gebruik: $DISPLAY_NAME $OPTIONS_USAGE_COMMON"
26: readonly HELP="Gebruik: $DISPLAY_NAME [OPTIE...]"
27:
28: Controlebestanden opsplitsen.
29:
30: Opties:
31: $OPTIONS_HELP_COMMON"
32:
33: readonly SEARCHDIR=$HOME/Downloads
34: readonly HASHPROG=sha256sum
35:
36:
37: #####
38: # Global variables
39: #####
40:
41: declare ISO_FILE=''
42: declare ISO_FILE_FOUND=false
43:
44: #####
45: # Functions
46: #####
47:
48: check_input() {
49:     local -i getopt_rc=0
50:     local parsed=''
51:
52:     parsed=$(
53:         getopt --alternative \
54:             --options "$OPTIONS_SHORT" \
55:             --longoptions "$OPTIONS_LONG" \
56:             --name "$DISPLAY_NAME" \
57:             -- "$@"
58:     ) || getopt_rc=$?
59:     if [[ $getopt_rc -ne $SUCCESS ]]; then
60:         printf '%s\n' "$USAGELINE" >&2
61:         exit $ERROR
62:     fi
63:     eval set -- "$parsed"
64:     process_common_options "$@"
65:
66:     while true; do
67:         case $1 in
68:             --)
```

```
69:         shift
70:         break
71:         ;;
72:     *)
73:         shift
74:         ;;
75:     esac
76: done
77:
78: if [[ "$*" ]]; then
79:     TEXT='geen argumenten opgeven'
80:     printf "$DISPLAY_NAME: %s\n%s\n" "$TEXT" "$USAGELINE" >&2
81:     exit $ERROR
82: fi
83:
84: # Een non-gui script gestart met optie gui.
85: if $OPTION_GUI; then
86:     OPTION_GUI=false
87:     DESKTOP_TERMINAL=false
88: fi
89:
90: check_user
91: }
92:
93:
94: process_input() {
95:     cd "$SEARCHDIR" || exit $ERROR
96:
97:     for ISO_FILE in *.iso; do
98:
99:         if ! [[ -f "$ISO_FILE" ]]; then
100:             continue
101:         fi
102:
103:         ISO_FILE_FOUND=true
104:
105:         create_checksum_file
106:
107:     done
108:
109:     if ! $ISO_FILE_FOUND; then
110:         warning "Geen beeldbestanden (.iso) gevonden in $SEARCHDIR."
111:         exit $WARNING
112:     fi
113: }
114:
115:
116: create_checksum_file() {
117:     local -a files=()
118:     local -i grep_rc=0
119:
120:     if [[ -e $ISO_FILE.$HASHPROG ]]; then
121:         info "Beeldbestand $ISO_FILE heeft al controlebestand \
122: $ISO_FILE.$HASHPROG.\n"
123:         return $SUCCESS
124:     else
125:         for file in *; do
126:             files+=("$file")
127:         done
128:         grep --word-regexp \
129:             --regexp="$ISO_FILE" \
130:             --regexp="*$ISO_FILE" \
131:             --exclude=*.iso \
132:             --exclude=*512* \
133:             --exclude-dir=* \
134:             --no-filename \
135:             -- "${files[@]}" \
136:             > "$ISO_FILE.$HASHPROG" 2> >($LOGCMD) || grep_rc=$?
```

```
137:         if [[ $grep_rc -ne $SUCCESS ]]; then
138:             error "Geen controlesom gevonden voor $ISO_FILE.\n"
139:         else
140:             info "Controlebestand $ISO_FILE.$HASHPROG aangemaakt.\n"
141:         fi
142:     fi
143: }
144:
145:
146: term_script() {
147:     rm --force --verbose -- *SHA* |& $LOGCMD
148:     if $ISO_FILE_FOUND; then
149:         info "Om de beeldbestanden (.iso) te controleren voer uit:
150:         ${BLUE}cd $SEARCHDIR;$HASHPROG --check --ignore-missing *.$HASHPROG; cd \
151:     -${NORMAL}
152:
153:
154: En om vervolgens USB-sticks te maken voer uit:
155:     ${BLUE}kz mkusbs${NORMAL}
156:
157:
158: Om de controlebestanden (.${HASHPROG}s) opnieuw te downloaden voer uit:
159:     ${BLUE}kz dliso --checksums${NORMAL}"
160:     fi
161:     exit $SUCCESS
162: }
163:
164:
165: #####
166: # Main line
167: #####
168:
169: main() {
170:     init_script "$@"
171:     check_input "$@"
172:     process_input
173:     term_script
174: }
175:
176:
177: main "$@"
178:
179:
180: # EOF
```