

```
1: #!/bin/bash
2: # shellcheck source=kz-common.sh
3: #####
4: # Pakket kz distribueren.
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-deploy
11: DISPLAY_NAME=${PROGRAM_NAME}/kz-/kz }
12: RELEASE_YEAR=2009
13:
14: VERSION_NUMBER=62.07.03
15: VERSION_DATE=2021-10-12
16:
17:
18: #####
19: # Global constants
20: #####
21:
22: readonly RUN_AS_SUPERUSER=false
23: readonly OPTIONS_SHORT=$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: Pakket kz distribueren.
29:
30: Opties:
31: $OPTIONS_HELP_COMMON"
32:
33: readonly DEB_REPO=$HOME/kz-deb
34: readonly DOCS_REPO=$HOME/kz-docs
35: readonly SCRIPTS_REPO=$HOME/kz-scripts
36: readonly UPLOADS_REPO=$HOME/kz-uploads
37: readonly UPLOADS_DISTDIR=$UPLOADS_REPO/dist
38:
39:
40: #####
41: # Global variables
42: #####
43:
44:
45: #####
46: # Functions
47: #####
48:
49: check_input() {
50:     local karel_full_name=''
51:     local ubuntu_full_name=''
52:     local debian_full_name=''
53:     local -i getopt_rc=0
54:     local parsed=''
55:     local prompt=''
56:
57:     parsed=$(
58:         getopt --alternative \
59:             --options "$OPTIONS_SHORT" \
60:             --longoptions "$OPTIONS_LONG" \
61:             --name "$DISPLAY_NAME" \
62:             -- "$@"
63:     ) || getopt_rc=$?
64:     if [[ $getopt_rc -ne $SUCCESS ]]; then
65:         printf '%s\n' "$USAGELINE" >&2
66:         exit $ERROR
67:     fi
68:     eval set -- "$parsed"
```

```
69: process_common_options "$@"
70:
71: while true; do
72:     case $1 in
73:         --)
74:             shift
75:             break
76:             ;;
77:         *)
78:             shift
79:             ;;
80:     esac
81: done
82:
83: if [[ "$*" ]]; then
84:     TEXT='geen argumenten opgeven'
85:     printf "$DISPLAY_NAME: %s\n%s\n" \
86:         "$TEXT" \
87:         "$USAGELINE" >&2
88:     exit $ERROR
89: fi
90:
91: karel_full_name=$(
92:     getent passwd karel |
93:     awk -F: '{print $5}' |
94:     awk -F, '{print $1}' || true
95: )
96: ubuntu_full_name=$(
97:     getent passwd ubuntu |
98:     awk -F: '{print $5}' |
99:     awk -F, '{print $1}' || true
100: )
101: debian_full_name=$(
102:     getent passwd user |
103:     awk -F: '{print $5}' |
104:     awk -F, '{print $1}' || true
105: )
106: if ! [[ ($USER = karel && $karel_full_name = 'Karel Zimmer' ) ||
107:         ($USER = ubuntu && $ubuntu_full_name = 'Live session user') ||
108:         ($USER = user && $debian_full_name = 'Debian Live user') ]]
109: then
110:     TEXT='alleen uitvoeren als ontwikkelaar'
111:     printf "$DISPLAY_NAME: %s\n%s\n" \
112:         "$TEXT" \
113:         "$USAGELINE" >&2
114:     exit $ERROR
115: fi
116:
117: # Een non-gui script gestart met optie gui.
118: if $OPTION_GUI; then
119:     OPTION_GUI=false
120:     DESKTOP_TERMINAL=false
121: fi
122:
123: check_user
124:
125: # Regel sudo maar gelijk voor verderop.
126: prompt="Authenticatie is vereist om delen van $DISPLAY_NAME uit te voeren.
127: [sudo] wachtwoord voor %p: "
128: sudo --prompt="$prompt" true
129: }
130:
131:
132: process_input() {
133:     check_dependencies
134:     check_names
135:     check_scripts
136:     build_package
```

```
137:     upload_website
138:     install_package
139: }
140:
141:
142: check_dependencies() {
143:     if ! command -v lftp 1> /dev/null; then
144:         # Voor upload website.
145:         info 'Installeer lftp...'
146:         sudo apt-get install --yes lftp |& $LOGCMD
147:     fi
148: }
149:
150:
151: check_names() {
152:     local -i check_rc=0
153:
154:     info 'Controleer bestandsnamen (kz ckname)...'
155:     log "Check filenames with 'kz ckname'."
156:     "$SCRIPTS_REPO"/kz ckname "$DEB_REPO" \
157:         "$DOCS_REPO" \
158:         "$SCRIPTS_REPO" || check_rc=$?
159:     if [[ $check_rc -ne $SUCCESS ]]; then
160:         warning "
161: Herstel alle meldingen hierboven en start daarna weer '$DISPLAY_NAME'."
162:         exit $check_rc
163:     fi
164: }
165:
166:
167: check_scripts() {
168:     local -i check_rc=0
169:
170:     info 'Controleer scripts (kz ivp)...'
171:     log "Check scripts with 'kz ivp'."
172:     "$SCRIPTS_REPO"/kz ivp || check_rc=$?
173:     if [[ $check_rc -ne $SUCCESS ]]; then
174:         warning "
175: Herstel alle meldingen hierboven en start daarna weer '$DISPLAY_NAME'."
176:         exit $check_rc
177:     fi
178: }
179:
180:
181: build_package() {
182:     local -i check_rc=0
183:
184:     info 'Bouw website en pakket (kz build)...'
185:     log "Build package with 'kz build'."
186:     "$SCRIPTS_REPO"/kz build || check_rc=$?
187:     if [[ $check_rc -ne $SUCCESS ]]; then
188:         warning "
189: Herstel alle meldingen hierboven en start daarna weer '$DISPLAY_NAME'."
190:         exit $check_rc
191:     fi
192: }
193:
194:
195: upload_website() {
196:     local ftp_set='set ssl:verify-certificate no'
197:     local ftp_from=$UPLOADS_DISTDIR
198:     local ftp_to=/httpdocs
199:     local ftp_opts='--reverse --delete --verbose'
200:     local ftp_cmd="mirror $ftp_opts $ftp_from $ftp_to; exit"
201:     local ftp_host=server106.hosting2go.nl
202:     local ftp_user=kzimmer
203:     local ftp_login=$HOME/.kz-$ftp_host
204:
```

```
205: info 'Upload website...'  
206: if ! [[ -f $ftp_login ]]; then  
207:   read -rsp "FTP-wachtwoord voor $ftp_host: "  
208:   printf '%s\n' "$REPLY" > "$ftp_login"  
209:   printf '\n'  
210:   chmod 'u=rw,g=,o=' "$ftp_login" |& $LOGCMD  
211: fi  
212: if ! lftp --user "$ftp_user" \\  
213:   --password "$(cat "$ftp_login")" \\  
214:   -e "$ftp_set; $ftp_cmd" \\  
215:   "$ftp_host" |& $LOGCMD; then  
216:   rm "$ftp_login"  
217:   warning 'Uploaden website is mislukt.'  
218:   warning 'Controleer de log...!'  
219:   exit $WARNING  
220: fi  
221: sleep 5  
222: }  
223:  
224:  
225: install_package() {  
226:   info 'Installeer pakket kz (kz getdeb)...'  
227:   # Constructie '2> >($LOGCMD)' om stderr naar de log te krijgen.  
228:   if ! wget -O- karelzimmer.nl/kz 2> >($LOGCMD) | bash; then  
229:     warning 'Installeren pakket kz is niet uitgevoerd.'  
230:     exit $WARNING  
231:   fi  
232: }  
233:  
234:  
235: term_script() {  
236:   exit $SUCCESS  
237: }  
238:  
239:  
240: #####  
241: # Main line  
242: #####  
243:  
244: main() {  
245:   init_script "$@"  
246:   check_input "$@"  
247:   process_input  
248:   term_script  
249: }  
250:  
251:  
252: main "$@"  
253:  
254:  
255: # EOF
```