

```
1: #!/bin/bash
2: # shellcheck source=kz-common.sh
3: #####
4: # Pakket kz bouwen.
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-build
11: DISPLAY_NAME=${PROGRAM_NAME}/kz-/kz }
12: RELEASE_YEAR=2021
13:
14: VERSION_NUMBER=01.01.00
15: VERSION_DATE=2021-10-15
16:
17:
18: #####
19: # Global constants
20: #####
21:
22: readonly RUN_AS_SUPERUSER=false
23: readonly OPTIONS_SHORT=$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: Pakket kz bouwen.
29:
30: Opties:
31: $OPTIONS_HELP_COMMON"
32:
33: readonly DOCS_REPO=$HOME/kz-docs
34: readonly SCRIPTS_REPO=$HOME/kz-scripts
35:
36: readonly DEB_REPO=$HOME/kz-deb
37: readonly DEB_APPDIR=$DEB_REPO/app
38: readonly DEB_DISTDIR=$DEB_REPO/dist
39: readonly DEB_NAME=kz_365_all.deb
40:
41: readonly UPLOADS_REPO=$HOME/kz-uploads
42: readonly UPLOADS_APPDIR=$UPLOADS_REPO/app
43: readonly UPLOADS_DISTDIR=$UPLOADS_REPO/dist
44:
45: readonly -A SOURCE_UPLOADS=(
46:     [APP]=$UPLOADS_APPDIR
47:     [DOCS]=$DOCS_REPO
48:     [SCRIPTS]=$SCRIPTS_REPO
49: )
50: readonly -A TARGET_UPLOADS=(
51:     [DIST]=$UPLOADS_DISTDIR
52:     [DEB]=$UPLOADS_DISTDIR/apps/kz
53:     [DOCS]=$UPLOADS_DISTDIR/data/linux/documents
54:     [PDFS]=$UPLOADS_DISTDIR/data/linux/pdfs
55:     [SCRIPTS]=$UPLOADS_DISTDIR/data/linux/scripts
56: )
57:
58: readonly -A SOURCE_DEB=(
59:     [APP]=$DEB_APPDIR
60:     [DOCS]=$DOCS_REPO
61:     [SCRIPTS]=$SCRIPTS_REPO
62: )
63: readonly -A TARGET_DEB=(
64:     [DIST]=$DEB_DISTDIR
65:     [APPLICATIONS]=$DEB_DISTDIR/usr/share/applications
66:     [BACKGROUNDS]=$DEB_DISTDIR/usr/share/backgrounds
67:     [BACKGROUND_PROP]=$DEB_DISTDIR/usr/share/gnome-background-properties
68:     [BIN]=$DEB_DISTDIR/usr/bin
```

```
69:         [DOC]=$DEB_DISTDIR/usr/share/doc/kz
70:         [MAN]=$DEB_DISTDIR/usr/share/man/man1
71:         [POLICY_1]=$DEB_DISTDIR/usr/share/polkit-1/actions
72:         [TABCOMPL]=$DEB_DISTDIR/usr/share/bash-completion/completions
73:     )
74:
75:
76: #####
77: # Global variables
78: #####
79:
80:
81: #####
82: # Functions
83: #####
84:
85: check_input() {
86:     local karel_full_name=''
87:     local ubuntu_full_name=''
88:     local debian_full_name=''
89:     local -i getopt_rc=0
90:     local parsed=''
91:
92:     parsed=$(
93:         getopt --alternative \
94:             --options "$OPTIONS_SHORT" \
95:             --longoptions "$OPTIONS_LONG" \
96:             --name "$DISPLAY_NAME" \
97:             -- "$@"
98:     ) || getopt_rc=$?
99:     if [[ $getopt_rc -ne $SUCCESS ]]; then
100:         printf '%s\n' "$USAGELINE" >&2
101:         exit $ERROR
102:     fi
103:     eval set -- "$parsed"
104:     process_common_options "$@"
105:
106:     while true; do
107:         case $1 in
108:             --)
109:                 shift
110:                 break
111:                 ;;
112:             *)
113:                 shift
114:                 ;;
115:         esac
116:     done
117:
118:     if [[ "$*" ]]; then
119:         TEXT='geen argumenten opgeven'
120:         printf "$DISPLAY_NAME: %s\n%s\n" \
121:             "$TEXT" \
122:             "$USAGELINE" >&2
123:         exit $ERROR
124:     fi
125:
126:     karel_full_name=$(
127:         getent passwd karel |
128:         awk -F: '{print $5}' |
129:         awk -F, '{print $1}' | true
130:     )
131:     ubuntu_full_name=$(
132:         getent passwd ubuntu |
133:         awk -F: '{print $5}' |
134:         awk -F, '{print $1}' | true
135:     )
136:     debian_full_name=$(
```

```
137: getent passwd user |
138: awk -F: '{print $5}' |
139: awk -F, '{print $1}' || true
140: )
141: if ! [[ ($USER = karel && $karel_full_name = 'Karel Zimmer' ) ||
142: ($USER = ubuntu && $ubuntu_full_name = 'Live session user') ||
143: ($USER = user && $debian_full_name = 'Debian Live user') ]]
144: then
145:     TEXT='alleen uitvoeren als ontwikkelaar'
146:     printf "$DISPLAY_NAME: %s\n%s\n" \
147:           "$TEXT" \
148:           "$USAGELINE" >&2
149:     exit $ERROR
150: fi
151:
152: # Een non-gui script gestart met optie gui.
153: if $OPTION_GUI; then
154:     OPTION_GUI=false
155:     DESKTOP_TERMINAL=false
156: fi
157:
158: check_user
159: }
160:
161:
162: process_input() {
163:     check_dependencies
164:     remove_backups
165:     # Er is een afhankelijkheid t.a.v. aanmaken .pdf, vandaar uploads eerst.
166:     build_uploads
167:     build_deb
168: }
169:
170:
171: check_dependencies() {
172:     if ! command -v encript 1> /dev/null; then
173:         # Voor <man-pag>.pdf.
174:         info 'Installeer encript...'
175:         sudo apt-get install --yes encript |& $LOGCMD
176:     fi
177:     if ! command -v fakeroot 1> /dev/null; then
178:         # Voor build .deb.
179:         info 'Installeer fakeroot...'
180:         sudo apt-get install --yes fakeroot |& $LOGCMD
181:     fi
182:     if ! command -v ghostscript 1> /dev/null; then
183:         # Voor <man-pag>.pdf (ps2pdf).
184:         info 'Installeer ghostscript...'
185:         sudo apt-get install --yes ghostscript |& $LOGCMD
186:     fi
187:     if ! command -v rsync 1> /dev/null; then
188:         # Voor build kz-deb en kz-uploads.
189:         info 'Installeer rsync...'
190:         sudo apt-get install --yes rsync |& $LOGCMD
191:     fi
192: }
193:
194:
195: remove_backups() {
196:     find "$UPLOADS_REPO" \
197:         -name '*~' -delete -or \
198:         -name '*.*~' -delete -or \
199:         -name '*.bak' -delete |& $LOGCMD
200: }
201:
202:
203: build_uploads() {
204:     local file=''
```

```
205: local changes=''
206:
207: info 'Bouw website...'
208:
209: # -----
210: # Vul dist/ met app/
211: # -----
212: cd "$UPLOADS_APPDIR" || exit $ERROR
213: while read -r file; do
214:     # Kopieer bestand als deze nieuw of gewijzigd is (deltabepaling).
215:     if ! diff "$file" "$UPLOADS_DISTDIR/$file" &> /dev/null; then
216:         cp --parents \
217:            --preserve \
218:            --verbose \
219:            "$file" \
220:            "$UPLOADS_DISTDIR" |& $LOGCMD
221:     fi
222: done <<(find . -type f -print)
223: cd "$HOME" || exit $ERROR
224:
225: # -----
226: # Vul dist/ met kz-docs
227: # -----
228: info 'Gewijzigde documenten:'
229: changes=false
230:
231: # Vanuit gedit aangemaakt.
232: rm --force "$HOME"/Documenten/*.pdf
233:
234: # Vanuit LibreOffice aangemaakt.
235: rm --force "$HOME"/Documenten/Checklists/*.pdf |& $LOGCMD
236:
237: cd "${SOURCE_UPLOADS[DOCS]}" || exit $ERROR
238: for file in *.odt *.txt; do
239:     # Kopieer bestand en maak pdf aan als deze nieuw of gewijzigd is
240:     # (deltabepaling).
241:     if ! diff "$file" "${TARGET_UPLOADS[DOCS]}/$file" &> /dev/null; then
242:         cp --preserve \
243:            --verbose \
244:            "$file" \
245:            "${TARGET_UPLOADS[DOCS]}" |& $LOGCMD
246:         lowriter --headless \
247:                 --convert-to pdf \
248:                 --outdir "${TARGET_UPLOADS[PDFS]}" \
249:                 "$file" |& $LOGCMD
250:         changes=true
251:         info " $file"
252:     fi
253: done
254: if ! $changes; then
255:     info ' Geen documenten gewijzigd.'
256: fi
257: cd "$HOME" || exit $ERROR
258:
259: # -----
260: # Vul dist/ met kz-scripts
261: # -----
262: info 'Gewijzigde scripts:'
263: changes=false
264:
265: cd "${SOURCE_UPLOADS[SCRIPTS]}" || exit $ERROR
266: for file in *; do
267:     if [[ $file = LICENSE || $file = README.md ]]; then
268:         continue
269:     elif [[ -d "$file" ]]; then
270:         continue
271:     fi
272:     # Kopieer bestand en maak pdf aan als deze nieuw of gewijzigd is
```

```

273: # (deltabepaling).
274: if ! diff "$file" "${TARGET_UPLOADS[SCRIPTS]}/$file" &> /dev/null; then
275:     cp --preserve \
276:        --verbose \
277:        "$file" \
278:        "${TARGET_UPLOADS[SCRIPTS]}" |& $LOGCMD
279: if grep --quiet --regexp='^'.TH ' "$file"; then
280:     # Man-pagina.
281:     man --troff \
282:         "${SOURCE_UPLOADS[SCRIPTS]}/$file" |
283:     ps2pdf - "${TARGET_UPLOADS[PDFS]}/$file.pdf" |& $LOGCMD
284: else
285:     # Script inclusief *.1, *.completion, *.desktop, en *.policy.
286:     enscript --quiet \
287:             --line-numbers \
288:             --title="$file" \
289:             --output=- \
290:             "$file" |
291:     ps2pdf - "${TARGET_UPLOADS[PDFS]}/$file.pdf" |& $LOGCMD
292: fi
293:     changes=true
294:     info " $file"
295: fi
296: done
297: if !$changes; then
298:     info ' Geen scripts gewijzigd.'
299: fi
300: cd "$HOME" || exit $ERROR
301: }
302: }
303:
304:
305: build_deb() {
306:     local basename=''
307:     local dir=''
308:     local file=''
309:     local newfile=''
310:     local prefix='nl.karelzimmer.'
311:
312:     info 'Bouw pakket... '
313:
314:     rsync --archive \
315:         --delete \
316:         --verbose \
317:         --exclude='LICENSE' \
318:         --exclude='README.md' \
319:         --exclude='.git*' \
320:         --delete-excluded \
321:         "${SOURCE_DEB[APP]}/" \
322:         "${TARGET_DEB[DIST]}" |& $LOGCMD
323:
324:     # Zorg dat de rechten goed staan voor de rsync.
325:     chmod 'u=rwx,g=rx,o=rx' -- "$SCRIPTS_REPO"/*
326:     chmod 'a-x' -- "$SCRIPTS_REPO"/*.* "$SCRIPTS_REPO"/LICENSE
327:
328:     # Zorg ervoor dat alle benodigde mappen beschikbaar zijn voor de rsync.
329:     for dir in "${TARGET_DEB[@]}"; do
330:         mkdir --parents "$dir"
331:     done
332:
333:     # Deze rsync is inclusief *.1, *.completion, *.desktop, en *.policy.
334:     rsync --archive \
335:         --delete \
336:         --verbose \
337:         --exclude='__pycache__' \
338:         --exclude='LICENSE' \
339:         --exclude='README.md' \
340:         --exclude='.git*' \

```

```

341:         --delete-excluded          \
342:         "${SOURCE_DEB[SCRIPTS]}/"  \
343:         "${TARGET_DEB[BIN]}"        |& $LOGCMD
344:
345: # Man-pagina's (.1)
346: rsync --archive                    \
347:       --delete                     \
348:       --verbose                     \
349:       --remove-source-files        \
350:       "${TARGET_DEB[BIN]}/*.1      \
351:       "${TARGET_DEB[MAN]}"          |& $LOGCMD
352: gzip --best --force "${TARGET_DEB[MAN]}/*.1 |& $LOGCMD
353:
354: # Tab-completion-scripts (.completion)
355: for file in "${TARGET_DEB[BIN]}/*.completion; do
356:     basename=$(basename "$file" .completion)
357:     mv "$file" "${TARGET_DEB[TABCOMPL]}/$basename" |& $LOGCMD
358: done
359:
360: # Bureaublad-configuratiebestanden (.desktop)
361: for file in "${TARGET_DEB[BIN]}/*.desktop; do
362:     basename=$(basename "$file")
363:     mv "$file" "${TARGET_DEB[APPLICATIONS]}/$basename" |& $LOGCMD
364: done
365:
366: # PolicyKit actiedefinitiebestanden (.policy)
367: for file in "${TARGET_DEB[BIN]}/*.policy; do
368:     basename=$(basename "$file")
369:     newfile="${TARGET_DEB[POLICY_1]}/$prefix$basename"
370:     mv "$file" "$newfile" |& $LOGCMD
371: done
372:
373: build_background_properties_file
374:
375: cp --archive                        \
376:    --update                         \
377:    "${TARGET_UPLOADS[PDFS]}/Checklist installatie"*.pdf \
378:    "${TARGET_DEB[DOC]}"              |& $LOGCMD
379:
380: fakeroot dpkg-deb                   \
381:          --build                     \
382:          "$DEB_DISTDIR"              \
383:          "${TARGET_UPLOADS[DEB]}/$DEB_NAME" |& $LOGCMD
384: }
385:
386:
387: build_background_properties_file() {
388:     local wallpaper=''
389:     local wallpapers_config_file=kz-wallpapers.xml
390:
391:     {
392:     printf '%s\n' '<?xml version="1.0" encoding="UTF-8"?>'
393:     printf '%s\n' '<!DOCTYPE wallpapers SYSTEM "gnome-wp-list.dtd">'
394:     printf '%s\n' '<wallpapers>'
395:     } > "${TARGET_DEB[BACKGROUND_PROP]}/$wallpapers_config_file"
396:
397:     cd "${TARGET_DEB[BACKGROUNDS]}" || exit $ERROR
398:     for wallpaper in *; do
399:         # Verwijder DEB_DISTDIR uit TARGET_DEB[BACKGROUNDS]. Voorbeeld:
400:         # /tmp/kz/usr/share/backgrounds/kz-gnome.png ->
401:         # /usr/share/backgrounds/kz-gnome.png
402:         filename=${TARGET_DEB[BACKGROUNDS]}/$wallpaper
403:         filename=${filename}/${DEB_DISTDIR}/
404:         printf '%s\n' "    <wallpaper>
405:         <name>${wallpaper%.*}</name>
406:         <filename>$filename</filename>
407:         <options>zoom</options>
408:         <pcolor>#000000</pcolor>

```

```
409:      <scolor>#000000</scolor>
410:      <shade_type>solid</shade_type>
411: </wallpaper>" \
412:     >> "${TARGET_DEB[BACKGROUND_PROP]}/$wallpapers_config_file"
413: done
414: printf '%s\n' \
415:     '</wallpapers>' >> "${TARGET_DEB[BACKGROUND_PROP]}/\
416: $wallpapers_config_file"
417: cd "$HOME" || exit $ERROR
418: }
419:
420:
421: term_script() {
422:     exit $SUCCESS
423: }
424:
425:
426: #####
427: # Main line
428: #####
429:
430: main() {
431:     init_script "$@"
432:     check_input "$@"
433:     process_input
434:     term_script
435: }
436:
437:
438: main "$@"
439:
440:
441: # EOF
```