

```
1: #!/bin/bash
2: # shellcheck source=kz-common.sh
3: #####
4: # GUI wijzigen.
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-gset
11: DISPLAY_NAME=${PROGRAM_NAME/kz-/kz }
12: RELEASE_YEAR=2015
13:
14: VERSION_NUMBER=12.00.09
15: VERSION_DATE=2021-10-04
16:
17:
18: #####
19: # Global constants
20: #####
21:
22: readonly RUN_AS_SUPERUSER=false
23: readonly OPTIONS_SHORT=$OPTIONS_SHORT_COMMON
24: readonly OPTIONS_LONG=$OPTIONS_LONG_COMMON",addappfolder,addfavbottom,\
25: addfavtop,removeappfolder,removefav,file:,folder:"
26: readonly USAGE="Gebruik: $DISPLAY_NAME \
27: [--addappfolder|--removeappfolder --folder=APPFOLDER]
28:         [--addfavbottom|--addfavtop|--removefav --file=DESKTOPFILE]
29:         $OPTIONS_USAGE_COMMON"
30: readonly HELP="Gebruik: $DISPLAY_NAME [OPTIE...]"
31:
32: GUI wijzigen.
33:
34: Opties:
35:     --addappfolder
36:         voeg een appfolder toe
37:     --addfavbottom
38:         voeg een toepassingspictogram (favoriet) toe onderaan aan het \
39: favorietengebied
40:     --addfavtop
41:         voeg een toepassingspictogram (favoriet) toe bovenaan aan het \
42: favorietengebied
43:     --file=DESKTOPFILE
44:         opgegeven desktopbestand (.desktop) verwerken
45:     --folder=APPFOLDER
46:         opgegeven appfolder verwerken
47:     --removeappfolder
48:         verwijder een appfolder
49:     --removefav
50:         verwijder een toepassingspictogrammen (favoriet) uit het \
51: favorietengebied
52: $OPTIONS_HELP_COMMON"
53:
54:
55: #####
56: # Global variables
57: #####
58:
59: declare CONFIG_A=''
60: declare CONFIG_B=''
61: declare DESKTOPFILE=''
62: declare FILE_ARGUMENT=''
63: declare FOLDER_ARGUMENT=''
64: declare OPTION_ADDAPPFOLDER=false
65: declare OPTION_ADDFAVBOTTOM=false
66: declare OPTION_ADDFAVTOP=false
67: declare OPTION_FILE=false
68: declare OPTION_FOLDER=false
```

```
69: declare OPTION_REMOVEAPPFOLDER=false
70: declare OPTION_REMOVEFAV=false
71:
72:
73: #####
74: # Functions
75: #####
76:
77: check_input() {
78:     local -i getopt_rc=0
79:     local     parsed=''
80:
81:     parsed=$(
82:         getopt --alternative           \
83:                --options              "$OPTIONS_SHORT" \
84:                --longoptions         "$OPTIONS_LONG"   \
85:                --name                 "$DISPLAY_NAME"  \
86:                --                     "$@"              \
87:     ) || getopt_rc=$?
88:     if [[ $getopt_rc -ne $SUCCESS ]]; then
89:         printf '%s\n' "$USAGELINE" >&2
90:         exit $ERROR
91:     fi
92:     eval set -- "$parsed"
93:     process_common_options "$@"
94:
95:     while true; do
96:         case $1 in
97:             --addappfolder)
98:                 if $OPTION_ADDAPPFOLDER; then
99:                     TEXT="optie '$1' eenmaal opgeven"
100:                    printf "$DISPLAY_NAME: %s\n%s\n" \
101:                           "$TEXT" \
102:                           "$USAGELINE" >&2
103:                    exit $ERROR
104:                else
105:                    OPTION_ADDAPPFOLDER=true
106:                fi
107:                 shift
108:             ;;
109:             --addfavtop)
110:                 if $OPTION_ADDFAVTOP; then
111:                     TEXT="optie '$1' eenmaal opgeven"
112:                    printf "$DISPLAY_NAME: %s\n%s\n" \
113:                           "$TEXT" \
114:                           "$USAGELINE" >&2
115:                    exit $ERROR
116:                 elif ($OPTION_ADDFAVBOTTOM || $OPTION_REMOVEFAV); then
117:                     TEXT="een optie eenmaal opgeven"
118:                    printf "$DISPLAY_NAME: %s\n%s\n" \
119:                           "$TEXT" \
120:                           "$USAGELINE" >&2
121:                    exit $ERROR
122:                 else
123:                     OPTION_ADDFAVTOP=true
124:                 fi
125:                 shift
126:             ;;
127:             --addfavbottom)
128:                 if $OPTION_ADDFAVBOTTOM; then
129:                     TEXT="optie '$1' eenmaal opgeven"
130:                    printf "$DISPLAY_NAME: %s\n%s\n" \
131:                           "$TEXT" \
132:                           "$USAGELINE" >&2
133:                    exit $ERROR
134:                 elif ($OPTION_ADDFAVTOP || $OPTION_REMOVEFAV); then
135:                     TEXT="een optie eenmaal opgeven"
136:                    printf "$DISPLAY_NAME: %s\n%s\n" \
```

```
137:             "$TEXT"                                \
138:             "$USAGELINE"                            >&2
139:             exit  $ERROR
140:         else
141:             OPTION_ADDFAVBOTTOM=true
142:         fi
143:         shift
144:         ;;
145:     --file)
146:         if $OPTION_FILE; then
147:             TEXT="optie '$1' eenmaal opgeven"
148:             printf "$DISPLAY_NAME: %s\n%s\n"      \
149:                 "$TEXT"                          \
150:                 "$USAGELINE"                    >&2
151:             exit  $ERROR
152:         else
153:             OPTION_FILE=true
154:             FILE_ARGUMENT=$2
155:         fi
156:         shift 2
157:         ;;
158:     --folder)
159:         if $OPTION_FOLDER; then
160:             TEXT="optie '$1' eenmaal opgeven"
161:             printf "$DISPLAY_NAME: %s\n%s\n"      \
162:                 "$TEXT"                          \
163:                 "$USAGELINE"                    >&2
164:             exit  $ERROR
165:         else
166:             OPTION_FOLDER=true
167:             FOLDER_ARGUMENT=$2
168:         fi
169:         shift 2
170:         ;;
171:     --removeappfolder)
172:         if $OPTION_REMOVEAPPFOLDER; then
173:             TEXT="optie '$1' eenmaal opgeven"
174:             printf "$DISPLAY_NAME: %s\n%s\n"      \
175:                 "$TEXT"                          \
176:                 "$USAGELINE"                    >&2
177:             exit  $ERROR
178:         else
179:             OPTION_REMOVEAPPFOLDER=true
180:         fi
181:         shift
182:         ;;
183:     --removefav)
184:         if $OPTION_REMOVEFAV; then
185:             TEXT="optie '$1' eenmaal opgeven"
186:             printf "$DISPLAY_NAME: %s\n%s\n"      \
187:                 "$TEXT"                          \
188:                 "$USAGELINE"                    >&2
189:             exit  $ERROR
190:         elif ($OPTION_ADDFAVTOP || $OPTION_ADDFAVBOTTOM); then
191:             TEXT="een optie eenmaal opgeven"
192:             printf "$DISPLAY_NAME: %s\n%s\n"      \
193:                 "$TEXT"                          \
194:                 "$USAGELINE"                    >&2
195:             exit  $ERROR
196:         else
197:             OPTION_REMOVEFAV=true
198:         fi
199:         shift
200:         ;;
201:     --)
202:         shift
203:         break
204:         ;;
```

```
205:         *)
206:             shift
207:             ;;
208:     esac
209: done
210:
211: if [[ "$*" ]]; then
212:     TEXT='geen argumenten opgeven'
213:     printf "$DISPLAY_NAME: %s\n%s\n" "$TEXT" "$USAGELINE" >&2
214:     exit $ERROR
215: fi
216:
217: if ($OPTION_ADDFAVTOP || $OPTION_ADDFAVBOTTOM || $OPTION_REMOVEFAV) &&
218:     ! $OPTION_FILE; then
219:     TEXT="ontbrekende optie 'file'"
220:     printf "$DISPLAY_NAME: %s\n%s\n" "$TEXT" "$USAGELINE" >&2
221:     exit $ERROR
222: elif $OPTION_FILE &&
223:     ! ($OPTION_ADDFAVTOP || $OPTION_ADDFAVBOTTOM || $OPTION_REMOVEFAV)
224:     then
225:     TEXT="ontbrekende optie 'addfavtop', 'addfavbottom', of 'removefav'"
226:     printf "$DISPLAY_NAME: %s\n%s\n" "$TEXT" "$USAGELINE" >&2
227:     exit $ERROR
228: elif $OPTION_FILE && [[ $FILE_ARGUMENT != *.desktop ]]; then
229:     TEXT="bestandsnaam dient te eindigen met '.desktop'"
230:     printf "$DISPLAY_NAME: %s\n%s\n" "$TEXT" "$USAGELINE" >&2
231:     exit $ERROR
232: fi
233: DESKTOPFILE=$FILE_ARGUMENT
234:
235: if ($OPTION_ADDAPPFOLDER || $OPTION_REMOVEAPPFOLDER) && ! $OPTION_FOLDER
236:     then
237:     TEXT="ontbrekende optie 'folder'"
238:     printf "$DISPLAY_NAME: %s\n%s\n" "$TEXT" "$USAGELINE" >&2
239:     exit $ERROR
240: elif $OPTION_FOLDER && ! ($OPTION_ADDAPPFOLDER || $OPTION_REMOVEAPPFOLDER)
241:     then
242:     TEXT="ontbrekende optie 'addappfolder'"
243:     printf "$DISPLAY_NAME: %s\n%s\n" "$TEXT" "$USAGELINE" >&2
244:     exit $ERROR
245: fi
246: APPFOLDER=$FOLDER_ARGUMENT
247:
248: # Een non-gui script gestart met optie gui.
249: if $OPTION_GUI; then
250:     OPTION_GUI=false
251:     DESKTOP_TERMINAL=false
252: fi
253:
254: check_user
255: }
256:
257:
258: process_input() {
259:     if $OPTION_ADDFAVTOP; then
260:         process_option_addfavtop
261:     elif $OPTION_ADDFAVBOTTOM; then
262:         process_option_addfavbottom
263:     elif $OPTION_REMOVEFAV; then
264:         process_option_removefav
265:     elif $OPTION_ADDAPPFOLDER; then
266:         process_option_addappfolder
267:     elif $OPTION_REMOVEAPPFOLDER; then
268:         process_option_removeappfolder
269:     else
270:         capture_gui_changes
271:     fi
272: }
```

```

273:
274:
275: process_option_addfavtop () {
276:     local favorite_apps=''
277:
278:     favorite_apps=$(gsettings get org.gnome.shell favorite-apps)
279:     if [[ $favorite_apps = '@as []' ]]; then
280:         gsettings set org.gnome.shell favorite-apps "['$DESKTOPFILE']"
281:     elif ! printf '%s' \
282:             "$favorite_apps" | \
283:             grep --quiet \
284:                 --word-regexp \
285:                 --regexp="$DESKTOPFILE"; then
286:         gsettings set \
287:             org.gnome.shell \
288:             favorite-apps \
289:             "['$DESKTOPFILE', $( \
290:                 printf '%s' \
291:                     "$favorite_apps" | \
292:                     awk -F[ \
293:                         '{print $2}' \
294:                     )"
295:     fi
296: }
297:
298:
299: process_option_addfavbottom () {
300:     local favorite_apps=''
301:
302:     favorite_apps=$(gsettings get org.gnome.shell favorite-apps)
303:     if [[ $favorite_apps = '@as []' ]]; then
304:         gsettings set org.gnome.shell favorite-apps "['$DESKTOPFILE']"
305:     elif ! printf '%s' \
306:             "$favorite_apps" | \
307:             grep --quiet \
308:                 --word-regexp \
309:                 --regexp="$DESKTOPFILE"; then
310:         gsettings set \
311:             org.gnome.shell \
312:             favorite-apps \
313:             "$( \
314:                 printf '%s' \
315:                     "$favorite_apps" | \
316:                     awk -F[ \
317:                         '{print $1}'), '$DESKTOPFILE']"
318:     fi
319: }
320:
321:
322: process_option_removefav () {
323:     local favorite_apps=''
324:
325:     favorite_apps=$(gsettings get org.gnome.shell favorite-apps)
326:     if [[ $favorite_apps = '@as []' ]]; then
327:         :
328:     elif [[ $favorite_apps = "['$DESKTOPFILE']" ]]; then
329:         gsettings set org.gnome.shell favorite-apps ""
330:     else gsettings set \
331:         org.gnome.shell \
332:         favorite-apps \
333:         "$( \
334:             printf '%s' \
335:                 "$favorite_apps" \
336:             sed --expression="s/'$DESKTOPFILE', //g" \
337:             sed --expression="s/, '$DESKTOPFILE' //" \
338:         )"
339:     fi
340: }

```

```
341:
342:
343: process_option_addappfolder () {
344:     local app_folders=''
345:     local schema=''
346:     local path=''
347:
348:     app_folders=$(gsettings get org.gnome.desktop.app-folders folder-children)
349:
350:     if [[ $app_folders = '@as []' ]]; then
351:         set org.gnome.desktop.app-folders folder-children ["'$APPFOLDER'"]
352:     elif ! printf '%s' \
353:             "$app_folders" | \
354:             grep --quiet \
355:                 --word-regexp \
356:                 --regexp="$APPFOLDER"; then
357:         gsettings set \
358:             org.gnome.desktop.app-folders \
359:             folder-children \
360:             "$(\
361:                 printf '%s' "$app_folders" | \
362:                 awk -F] '{print $1}'), '$APPFOLDER'"
363:     fi
364:     schema=org.gnome.desktop.app-folders.folder
365:     path=/org/gnome/desktop/app-folders/folders/$APPFOLDER/
366:     gsettings set "$schema":"$path" name "$APPFOLDER"
367:     gsettings set "$schema":"$path" translate true
368:     gsettings set "$schema":"$path" categories ["'$APPFOLDER'"]
369: }
370:
371:
372: process_option_removeappfolder () {
373:     local app_folders=''
374:     local schema=''
375:     local path=''
376:
377:     app_folders=$(gsettings get org.gnome.desktop.app-folders folder-children)
378:
379:     if [[ $app_folders = '@as []' ]]; then
380:         :
381:     elif [[ $app_folders = ["'$APPFOLDER'"] ]]; then
382:         gsettings set org.gnome.desktop.app-folders folder-children []
383:     else gsettings set \
384:         org.gnome.desktop.app-folders \
385:         folder-children \
386:         "$(\
387:             printf '%s' "$app_folders" | \
388:             sed --expression="s/, '$APPFOLDER'//"
389:         )"
390:     fi
391:     schema=org.gnome.desktop.app-folders.folder
392:     path=/org/gnome/desktop/app-folders/folders/$APPFOLDER/
393:     gsettings reset-recursively "$schema":"$path"
394: }
395:
396:
397: capture_gui_changes() {
398:     CONFIG_A=$(mktemp -t "$PROGRAM_NAME-A-XXXXXXXXXX.lst")
399:     save_configuration_database A "$CONFIG_A"
400:     info 'Voer nu de wijziging uit in de grafische werkomgeving.'
401:     read -rp 'Druk op de Enter-toets wanneer gereed [Enter]: '
402:     CONFIG_B=$(mktemp -t "$PROGRAM_NAME-B-XXXXXXXXXX.lst")
403:     save_configuration_database B "$CONFIG_B"
404:     report_database_changes "$CONFIG_A" "$CONFIG_B"
405: }
406:
407:
408: save_configuration_database() {
```

```
409: local fase=${1:-fase?}
410: local output_file=${2:-output_file?}
411:
412:
413: gsettings list-recursively > "$output_file" 2> >($LOGCMD)
414:
415: sort --unique \
416:      --output="$output_file" \
417:      "$output_file"
418:
419: info "Inhoud configuratiedatabase vastgelegd ($fase)."
```

```
420: }
421:
422:
423: report_database_changes() {
424:     local output_file1=${1:-output_file1?}
425:     local output_file2=${2:-output_file2?}
426:
427:     info '
428: WIJZIGINGen in de configuratiedatabase,
429: < is de oude instelling (A),
430: > is de nieuwe instelling (B):'
431:     diff "$output_file1" "$output_file2" | grep --regexp='[>|<]' || true
432:     info "
433: Bij WIJZIGINGen voer uit:
434:   ${BLUE}gsettings set WIJZIGING${NORMAL}"
435:     rm "$CONFIG_A" "$CONFIG_B"
436: }
437:
438:
439: term_script() {
440:     exit $SUCCESS
441: }
442:
443:
444: #####
445: # Main line
446: #####
447:
448: main() {
449:     init_script "$@"
450:     check_input "$@"
451:     process_input
452:     term_script
453: }
454:
455:
456: main "$@"
457:
458:
459: # EOF
```