

```
1: # shellcheck shell=bash
2: #####
3: # Algemene module voor shell scripts.
4: #
5: # Geschreven door Karel Zimmer <info@karelzimmer.nl>.
6: #####
7: PROGRAM_NAME=kz-common.sh
8: DISPLAY_NAME=${PROGRAM_NAME/kz-/kz }
9: RELEASE_YEAR=2009
10:
11: VERSION_NUMBER=27.03.03
12: VERSION_DATE=2021-10-07
13:
14:
15: #####
16: # Common global constants
17: #####
18:
19: readonly SUCCESS=0
20: readonly ERROR=1
21: readonly WARNING=2
22:
23: # shellcheck disable=SC2034
24: readonly OPTIONS_SHORT_COMMON='dghuv'
25: # shellcheck disable=SC2034
26: readonly OPTIONS_LONG_COMMON='debug,gui,help,usage,version'
27: # shellcheck disable=SC2034
28: readonly OPTIONS_USAGE_COMMON="[-u|--usage] [-h|--help] [-v|--version] \
29: [-g|--gui] [-d|--debug]"
30: # shellcheck disable=SC2034
31: readonly OPTIONS_HELP_COMMON="\
32: -h --help      toon deze hulptekst
33: -u --usage     toon een korte gebruikssamenvatting
34: -v --version   toon programmaversie
35: -g --gui       start in grafische modus indien van toepassing
36: -d --debug     neem foutopsporingsinformatie op in het logboek"
37:
38: DISTRO=$(lsb_release --id --short | tr '[:upper:]' '[:lower:]')
39: readonly DISTRO
40: THIS_YEAR=$(date +%Y); readonly THIS_YEAR
41:
42:
43: #####
44: # Common global variables
45: #####
46:
47: declare -a CMDLINE_ARGS=()
48: declare     BACKUP_TO_DELETE=''
49: declare     HELP='Gebruik: source kz-common.sh
50:           of: . kz-common.sh'
51: declare     LOGCMD=''
52: declare     LOGCMD_CHECK=''
53: declare     OPTION_DEBUG=false
54: declare     OPTION_GUI=false
55: declare     OPTION_HELP=false
56: declare     OPTION_USAGE=false
57: declare     OPTION_VERSION=false
58: declare     RUN_AS_SUPERUSER=false
59: declare     DESKTOP_TERMINAL=false
60: declare     TEXT=''
61: declare     TITLE=''
62: declare     USAGE='Gebruik: source kz-common.sh
63:           of: . kz-common.sh'
64: declare     USAGELINE=''
65:
66: # Terminalattributen, zie 'man terminfo'. Gebruik ${<variabele-naam>}.
67: declare     NORMAL=''
68: declare     BOLD=''
```

```
69: declare RED=''
70: declare GREEN=''
71: declare YELLOW=''
72: declare BLUE=''
73:
74:
75: #####
76: # Common functions
77: #####
78:
79: check_on_ac_power() {
80:     local -i on_battery=0
81:
82:     on_ac_power >/dev/null 2>&1 || on_battery=$?
83:     if [[ on_battery -eq 1 ]]; then
84:         warning '
85: De computer gebruikt nu alleen de accu voor de stroomvoorziening.
86:
87: Geadviseerd wordt om de computer aan te sluiten op het stopcontact.
88: '
89:     fi
90: }
91:
92:
93: error() {
94:     if $OPTION_GUI; then
95:         TITLE="Foutmelding $DISPLAY_NAME"
96:         # Constructie '2> >($LOGCMD)' om stderr naar de log te krijgen.
97:         # Voorbeeld: Unable to init server: Kon niet verbinden:
98:         #             Verbinding is geweigerd
99:         #             en: (zenity:47712): Gtk-WARNING **: 10:35:49.339:
100:        #             cannot open display:
101:        if ! zenity --error \
102:            --no-markup \
103:            --width 500 \
104:            --height 100 \
105:            --title "$TITLE" \
106:            --text "$@" \
107:            --ok-label 'OkÃ©' 2> >($LOGCMD); then
108:            true
109:        fi
110:    else
111:        printf "${RED}%b${NORMAL}\n" "$@"
112:    fi
113: }
114:
115:
116: check_user() {
117:     local prompt=''
118:
119:     if $RUN_AS_SUPERUSER; then
120:         if [[ $UID -ne 0 ]]; then
121:             log "Restarted (exec sudo $0 ${CMDLINE_ARGS[*]})." --priority=debug
122:             prompt="Authenticatie is vereist om $DISPLAY_NAME uit te voeren.
123: [sudo] wachtwoord voor %p: "
124:             exec sudo --prompt="$prompt" "$0" "${CMDLINE_ARGS[@]}"
125:         fi
126:     else
127:         if [[ $UID -eq 0 ]]; then
128:             info "$DISPLAY_NAME: niet uitvoeren met 'sudo' of als root"
129:             exit $WARNING
130:         fi
131:     fi
132: }
133:
134:
135: info() {
136:     if $OPTION_GUI; then
```

```
137: TITLE="Informatie $DISPLAY_NAME"
138: # Constructie '2> >($LOGCMD)' om stderr naar de log te krijgen.
139: # Voorbeeld: Unable to init server: Kon niet verbinden:
140: #           Verbinding is geweigerd
141: #           en: (zenity:47712): Gtk-WARNING **: 10:35:49.339:
142: #           cannot open display:
143: if ! zenity --info \
144:           --no-markup \
145:           --width 600 \
146:           --height 100 \
147:           --title "$TITLE" \
148:           --text "$@" \
149:           --ok-label 'OkÃ@' 2> >($LOGCMD); then
150:     true
151: fi
152: else
153:     printf '%b\n' "$@"
154: fi
155: }
156:
157:
158: init_script() {
159:     # Script-hardening.
160:     set -o errexit
161:     set -o errtrace
162:     set -o nounset
163:     set -o pipefail
164:     # bash 5.0: trap exit: ${FUNCNAME[0]}: ongebonden variabele
165:     # Enable code-stepping:
166:     # trap '(read -p "[${BASH_SOURCE:$LINENO} ${BASH_COMMAND}?)"' DEBUG
167:     trap 'signal error $LINENO ${FUNCNAME:--} "${BASH_COMMAND} $?" ERR
168:     trap 'signal exit $LINENO ${FUNCNAME:--} "${BASH_COMMAND} $?" EXIT
169:     trap 'signal sighup $LINENO ${FUNCNAME:--} "${BASH_COMMAND} $?" SIGHUP # 1
170:     trap 'signal sigint $LINENO ${FUNCNAME:--} "${BASH_COMMAND} $?" SIGINT # 2
171:     trap 'signal sigterm $LINENO ${FUNCNAME:--} "${BASH_COMMAND} $?" SIGTERM #15
172:
173:     LOGCMD="systemd-cat --identifier=$PROGRAM_NAME"
174:     LOGCMD_CHECK="[sudo] journalctl --all --no-pager \
175: --identifier=$PROGRAM_NAME --since='$(date '+%Y-%m-%d %H:%M:%S')'"
176:
177:     CMDLINE_ARGS=("$@")
178:     log "Started (as $USER $0 ${CMDLINE_ARGS[*]} from $PWD)." --priority=notice
179:     if [[ $DISTRO = 'debian' && $UID -ne 0 ]]; then
180:         log '(++) Met Debian heeft gebruiker root toegang nodig
181: (++) tot mijn X-sessie voor het kunnen gebruiken van
182: (++) zenity in kz-scripts met RUN_AS_SUPERUSER=true:
183: (++) xhost +si:localuser:root' --priority=debug
184:         xhost +si:localuser:root |& $LOGCMD --priority=debug
185:     fi
186:
187:     # shellcheck disable=SC2034
188:     USAGELINE="Typ '$DISPLAY_NAME --usage' voor meer informatie."
189: }
190:
191:
192: log() {
193:     printf '%b' "$1" |& $LOGCMD "${2:---priority=info}"
194: }
195:
196:
197: process_common_options() {
198:     while true; do
199:         case $1 in
200:             -d|--debug)
201:                 OPTION_DEBUG=true
202:                 shift
203:                 ;;
204:             -u|--usage)
```

```
205:         OPTION_USAGE=true
206:         shift
207:         ;;
208:     -h|--help)
209:         OPTION_HELP=true
210:         shift
211:         ;;
212:     -v|--version)
213:         OPTION_VERSION=true
214:         shift
215:         ;;
216:     -g|--gui)
217:         OPTION_GUI=true
218:         shift
219:         ;;
220:     --)
221:         break
222:         ;;
223:     *)
224:         shift
225:         ;;
226:     esac
227: done
228:
229: if ! $OPTION_GUI && [[ -t 1 ]]; then
230:     # Tekstuitvoer non-gui naar de terminal.
231:     NORMAL=$(tput sgr0)
232:     BOLD=$(tput bold)
233:     RED=${BOLD}$(tput setaf 1)
234:     GREEN=${BOLD}$(tput setaf 2)
235:     YELLOW=${BOLD}$(tput setaf 3)
236:     BLUE=${BOLD}$(tput setaf 4)
237: fi
238:
239: if $OPTION_DEBUG; then
240:     process_option_debug
241: fi
242: if $OPTION_HELP; then
243:     process_option_help
244:     exit $SUCCESS
245: elif $OPTION_USAGE; then
246:     process_option_usage
247:     exit $SUCCESS
248: elif $OPTION_VERSION; then
249:     process_option_version
250:     exit $SUCCESS
251: fi
252: }
253:
254:
255: process_option_debug() {
256:     # Enable code-stepping.
257:     # trap '(read -p "[${BASH_SOURCE:$LINENO}] $BASH_COMMAND?)"' DEBUG
258:     printf "${YELLOW}%s\n${NORMAL}" '*** START DEBUG-SESSIE ***'
259:     log 'START DEBUG-SESSIE' --priority=debug
260:     log 'Start show current environment' --priority=debug
261:     log 'uname --all:' --priority=debug
262:     log "$(uname --all)" --priority=debug
263:     log 'lsb_release --all:' --priority=debug
264:     log "$(lsb_release --all 2> >($LOGCMD --priority=debug))"
265:     log 'cat /etc/os-release:' --priority=debug
266:     log "$(cat /etc/os-release)" --priority=debug
267:     log 'declare -p:' --priority=debug
268:     log "$(declare -p)" --priority=debug
269:     log 'End show current environment' --priority=debug
270:     log 'Routing bash xtrace to FD4' --priority=debug
271:     exec 4> >($LOGCMD --priority=debug)
272:     BASH_XTRACEFD=4
```

```
273: log 'Setting bash options'
274: set -o verbose
275: set -o xtrace
276: log 'Let the fun begin...' --priority=debug
277: # shellcheck disable=SC2016
278: ps4='+(${BASH_SOURCE}:${LINENO}): ${FUNCNAME[0]:+${FUNCNAME[0]}}(): {'
279: export PS4=$ps4
280: unset ps4
281: }
282:
283:
284: process_option_help() {
285:     printf '%s\n\n%s\n' \
286:         "$HELP" \
287:         "Typ 'man $DISPLAY_NAME' voor meer informatie."
288: }
289:
290:
291: process_option_usage() {
292:     printf '%s\n\n%s\n' \
293:         "$USAGE" \
294:         "Typ '$DISPLAY_NAME --help' voor meer informatie."
295: }
296:
297:
298: process_option_version() {
299:     local copyright_years='1970'
300:
301:     if [[ $RELEASE_YEAR -eq $THIS_YEAR ]]; then
302:         copyright_years=$RELEASE_YEAR
303:     else
304:         copyright_years=$RELEASE_YEAR-$THIS_YEAR
305:     fi
306:     info "$DISPLAY_NAME $VERSION_NUMBER ($VERSION_DATE)
307:
308: Geschreven door Karel Zimmer <info@karelzimmer.nl>.
309:
310: Auteursrecht (c) $copyright_years Karel Zimmer.
311: GNU Algemene Publieke Licentie <https://www.gnu.org/licenses/gpl.html>."
312: }
313:
314:
315: signal() {
316:     local signal=${1:-signal?}
317:     local -i lineno=${2:-0}
318:     local function=${3:-funcname?}
319:     local command=${4:-command?}
320:     local -i rc=${5:-1}
321:     local rc_desc=''
322:     local -i rc_desc_signalno=0
323:     local status="${RED}$ERROR/ERROR${NORMAL}"
324:
325:     case $rc in
326:         0)
327:             rc_desc='successful termination'
328:             status="${GREEN}$rc/SUCCESS${NORMAL}"
329:             ;;
330:         1)
331:             rc_desc='terminated with error'
332:             ;;
333:         2)
334:             rc_desc='terminated with warning'
335:             status="${YELLOW}$rc/WARNING${NORMAL}"
336:             ;;
337:         6[4-9]|7[0-8]) # 64 - 78
338:             rc_desc="open file '/usr/include/sysexit.h' and look for '$rc'"
339:             ;;
340:         126)
```

```
341:         rc_desc='command cannot execute'
342:         ;;
343:     127)
344:         rc_desc='command not found'
345:         ;;
346:     128)
347:         rc_desc='invalid argument to exit'
348:         ;;
349:     129)
350:         rc_desc='hangup'
351:         ;;
352:     130)
353:         rc_desc='terminated by control-c'
354:         ;;
355:     13[1-9]|14[0-2])
356:         rc_desc_signalno=$((rc - 128))
357:         rc_desc="typ 'trap -l' and look for '$rc_desc_signalno)'"
358:         ;;
359:     143)
360:         rc_desc='termination signal'
361:         ;;
362:     14[4-9]|1[5-8][0-9]|19[0-2])
363:         rc_desc_signalno=$((rc - 128))
364:         rc_desc="typ 'trap -l' and look for '$rc_desc_signalno)'"
365:         ;;
366:     255)
367:         rc_desc='exit status out of range'
368:         ;;
369:     *)
370:         rc_desc='unknown error'
371:         ;;
372: esac
373:
374:     log "signal: $signal, line: $lineno, function: $function, command: \
375: $command, code: $rc ($rc_desc)" --priority=debug
376:
377:     case $signal in
378:         error)
379:             error "Opdracht is afgebroken."
380:             exit "$rc"
381:             ;;
382:         exit)
383:             signal_exit
384:             if $OPTION_DEBUG; then
385:                 set +o verbose
386:                 set +o xtrace
387:                 BASH_XTRACEFD=''
388:                 exec 4>&-
389:                 printf "${YELLOW}%s\n${NORMAL}" '*** EINDE DEBUG-SESSIE ***'
390:                 printf '%s\n' "Opdracht om de log uit te lezen:
391: ${BLUE}$LOGCMD_CHECK${NORMAL}"
392:                 log 'EINDE DEBUG-SESSIE'
393:             fi
394:             log "Ended (code=exited, status=$status)." --priority=notice
395:             # Een non-gui script gestart met optie gui.
396:             if $DESKTOP_TERMINAL; then
397:                 TEXT="Druk op de Enter-toets om verder te gaan [Enter]: "
398:                 # < /dev/tty voor als FD 1 al in gebruik is.
399:                 read -rp "$TEXT" < /dev/tty
400:             fi
401:             trap - ERR EXIT SIGINT SIGPIPE SIGTERM SIGHUP
402:             exit "$rc"
403:             ;;
404:     *)
405:         warning "Opdracht is onderbroken."
406:         exit "$rc"
407:         ;;
408: esac
```

```
409: }
410:
411:
412: signal_exit() {
413:     case $PROGRAM_NAME in
414:         kz-backup)
415:             if [[ -e $BACKUP_TO_DELETE ]]; then
416:                 info 'Programma kz-backup is onderbroken.'
417:
418:                 Opgeslagen back-up wordt verwijderd...'
419:                 rm "$BACKUP_TO_DELETE" |& $LOGCMD
420:                 info 'Opgeslagen back-up is verwijderd.'
421:             fi
422:             ;;
423:         kz-getdeb)
424:             rm --force /tmp/"$PROGRAM_NAME"-?????????.deb /tmp/kz-common.sh
425:             # Script kz-getdeb verwijdert niet kz en kz.1 ivm lokale Git-repo.
426:             rm --force kz.{2..99}
427:             # Maar wel als in HOME, zoals voorgeschreven.
428:             cd "$HOME" || exit $ERROR
429:             rm --force kz kz.1
430:             if [[ $rc -ne $SUCCESS ]]; then
431:                 log "Als de pakketbeheerder 'apt' foutmeldingen geeft, start \
432: dan een Terminalvenster, en voer uit:
433: ${BLUE}sudo dpkg --configure --pending${NORMAL}
434: ${BLUE}sudo apt-get update --fix-missing${NORMAL}
435: ${BLUE}sudo apt-get install --fix-broken${NORMAL}
436: ${BLUE}sudo update-initramfs -u${NORMAL}" --priority=debug
437:             fi
438:             ;;
439:         kz-install)
440:             printf '%s' "${NORMAL}${CURSOR_VISIBLE}"
441:             rm --force /tmp/"$PROGRAM_NAME"-?????????.*
442:             if [[ $rc -ne $SUCCESS ]]; then
443:                 log "Als de pakketbeheerder 'apt' foutmeldingen geeft, start \
444: dan een Terminalvenster, en voer uit:
445: ${BLUE}sudo dpkg --configure --pending${NORMAL}
446: ${BLUE}sudo apt-get update --fix-missing${NORMAL}
447: ${BLUE}sudo apt-get install --fix-broken${NORMAL}
448: ${BLUE}sudo update-initramfs -u${NORMAL}" --priority=debug
449:             fi
450:             ;;
451:         kz-setup)
452:             printf '%s' "${NORMAL}${CURSOR_VISIBLE}"
453:             rm --force /tmp/"$PROGRAM_NAME"-?????????.*
454:             ;;
455:     *)
456:         return $SUCCESS
457:     ;;
458: esac
459: }
460:
461:
462: warning() {
463:     if $OPTION_GUI; then
464:         TITLE="Waarschuwing $DISPLAY_NAME"
465:         # Constructie '2> >($LOGCMD)' om stderr naar de log te krijgen.
466:         # Voorbeeld: Unable to init server: Kon niet verbinden:
467:         #             Verbinding is geweigerd
468:         #             en: (zenity:47712): Gtk-WARNING **: 10:35:49.339:
469:         #             cannot open display:
470:         if ! zenity --warning \
471:             --no-markup \
472:             --width 500 \
473:             --height 100 \
474:             --title "$TITLE" \
475:             --text "$@" \
476:             --ok-label 'OkÃ©' 2> >($LOGCMD); then
```

```
477:         true
478:     fi
479: else
480:     printf "${YELLOW}%b${NORMAL}\n" "$@" >&2
481: fi
482: }
483:
484:
485: # EOF
```