

```
1: # shellcheck shell=bash
2: # -----
3: # Scripts algemeen.
4: #
5: # Geschreven door Karel Zimmer <info@karelzimmer.nl>.
6: #
7: # Auteursrecht (c) 2009-2021 Karel Zimmer.
8: # GNU Algemene Publieke Licentie <https://www.gnu.org/licenses/gpl.html>.
9: #
10: # Kopieer dit bestand ("library file") in een script ("include").
11: # Dit om te voldoen aan de zogenaamde "SPOT rule" (Single Point of Truth
12: # regel): alle scripts gebruiken dezelfde gemeenschappelijke variabelen en
13: # functies.
14: # Toelichting op de versienummering:
15: #   VV.RR.MM   Versie.Release.Modificatie
16: #   ==.==.==   Alleen commentaar of stijl gewijzigd (geen code gewijzigd)
17: #   ==.==.+1   Kleine niet-functionele wijziging (patch/bugfix)
18: #   ==.+1.00   Kleine functionele wijziging (minor), opnemen in Logboek Linux
19: #   +1.00.00   Nieuwe functionaliteit (major), opnemen in Logboek Linux
20: #
21: # RelNum=26.14.03
22: # RelDat=2021-01-21
23: # -----
24:
25: # -----
26: # Common global constants
27: # -----
28: declare -ir SUCCESS=0
29: declare -ir ERROR=1
30: # shellcheck disable=SC2034
31: declare -ir WARNING=2
32:
33: readonly CALLED=$(printf '%s' "$0 $" | xargs --null)
34: # shellcheck disable=SC2034
35: readonly DASHES=$(printf '%.0s-' {1..79})
36: readonly PROGDIR=$(cd "$(dirname "$0")" && pwd)
37: readonly PROGRAMNAME=$(basename "$0")
38:
39:
40: # Bij aanpassingen ook .completion aanpassen!
41: # shellcheck disable=SC2034
42: readonly OPTIONS_SHORT_COMMON='dghuv'
43: # shellcheck disable=SC2034
44: readonly OPTIONS_LONG_COMMON='debug,gui,help,usage,version'
45: # shellcheck disable=SC2034
46: readonly OPTIONS_USAGE_COMMON="[-d|--debug] [-g|--gui] [-h|--help] \
47: [-u|--usage] [-v|--version] [--]"
48: # shellcheck disable=SC2034
49: readonly OPTIONS_HELP_COMMON="\
50: -d --debug      geef foutopsporingsinformatie weer in het logboek
51: -g --gui        start in grafische modus
52: -h --help       deze hulptekst tonen
53: -u --usage      een korte gebruikssamenvatting tonen
54: -v --version    de versie tonen
55: --             stop verwerken opdrachtregelopties"
56:
57: # Terminalattributen ('man terminfo'). Gebruik ${<variabele_naam>}.
58: readonly BOLD=$(tput bold)
59: readonly NORMAL=$(tput sgr0)
60: readonly RED=${BOLD}${tput setaf 1}
61: readonly GREEN=${BOLD}${tput setaf 2}
62: readonly YELLOW=${BOLD}${tput setaf 3}
63: readonly BLUE=${BOLD}${tput setaf 4}
64: readonly CURSOR_VISIBLE=$(tput cvvis)
65: # shellcheck disable=SC2034
66: readonly ERASE_LINE="$(tput el)\c"
67: # shellcheck disable=SC2034
68: readonly UP_ONE_LINE=$(tput cuul)
```

```
69:
70: # -----
71: # Common global variables
72: # -----
73: declare BACKUP_TO_DELETE=''
74: declare CHKLOGCMD=''
75: declare LESS_OPTIONS=''
76: declare LOGCMD=''
77: declare LOGCMD_DEBUG=''
78: declare OPTION_DEBUG=false
79: declare OPTION_GUI=false
80: declare OPTION_HELP=false
81: declare OPTION_USAGE=false
82: declare OPTION_VERSION=false
83: declare RUN_AS_SUPERUSER=false
84: declare TERMINAL=false
85: declare USAGELINE=''
86:
87: # -----
88: # Common functions
89: # -----
90: check_user() {
91:     if $RUN_AS_SUPERUSER; then
92:         if [[ $UID -ne 0 ]]; then
93:             log "restart w/ exec sudo $CALLED"
94:             # shellcheck disable=SC2086
95:             exec sudo $CALLED
96:         fi
97:     else
98:         if [[ $UID -eq 0 ]]; then
99:             info "$PROGNAME: niet uitvoeren met 'sudo' of als root"
100:            exit $ERROR
101:        fi
102:    fi
103: }
104:
105: check_on_ac_power() {
106:     local -i on_battery=0
107:
108:     on_ac_power >/dev/null 2>&1 || on_battery=$?
109:     if [[ on_battery -eq 1 ]]; then
110:         warning '
111: De computer gebruikt nu alleen de accu voor de stroomvoorziening.
112:
113: Geadviseerd wordt om de computer aan te sluiten op het stopcontact.
114: '
115:     fi
116: }
117:
118: error() {
119:     local text=${1:-text?}
120:
121:     if $OPTION_GUI; then
122:         # Constructie '2> >($LOGCMD)' om stderr naar de log te krijgen.
123:         # Voorbeeld: Unable to init server: Kon niet verbinden:
124:         #             Verbinding is geweigerd
125:         #             en: (zenity:47712): Gtk-WARNING **: 10:35:49.339:
126:         #             cannot open display:
127:         if ! zenity --error \
128:             --no-markup \
129:             --width      500 \
130:             --height    100 \
131:             --title     "Foutmelding $PROGNAME" \
132:             --text      "$text" \
133:             --ok-label  'OkÃ©' \
134:             2> >($LOGCMD); then
135:             true
136:         fi
```

```
137:     else
138:         printf "${RED}%b${NORMAL}\n" "$text"
139:     fi
140: }
141:
142: info() {
143:     local text=${1-text?}
144:
145:     if $OPTION_GUI; then
146:         # Constructie '2> >($LOGCMD)' om stderr naar de log te krijgen.
147:         # Voorbeeld: Unable to init server: Kon niet verbinden:
148:         #             Verbinding is geweigerd
149:         #             en: (zenity:47712): Gtk-WARNING **: 10:35:49.339:
150:         #             cannot open display:
151:         if ! zenity --info                                \
152:                 --no-markup                             \
153:                 --width 600                             \
154:                 --height 100                           \
155:                 --title "Informatie $PROGNAME"         \
156:                 --text "$text"                         \
157:                 --ok-label 'OkÃ©'                      \
158:                 2> >($LOGCMD); then
159:             true
160:         fi
161:     else
162:         printf '%b\n' "$text"
163:     fi
164: }
165:
166: init_script() {
167: # Enable code-stepping.
168: # trap '(read -p "[${BASH_SOURCE:$LINENO} $BASH_COMMAND?)"' DEBUG
169:     if command -v systemd && /dev/null; then
170:         LOGCMD="systemd-cat --identifier=$PROGNAME --priority=info"
171:         LOGCMD_DEBUG="systemd-cat --identifier=$PROGNAME --priority=debug"
172:         CHKLOGCMD="journalctl --all --no-pager --identifier=$PROGNAME \
173: --since='$(date '+%Y-%m-%d %H:%M:%S')'"
174:     else
175:         LOGCMD="logger --tag $PROGNAME --priority info"
176:         LOGCMD_DEBUG="logger --tag $PROGNAME --priority debug"
177:         CHKLOGCMD="grep $PROGNAME /var/log/syslog"
178:     fi
179:     log "started as $CALLED (from $PWD)"
180:
181:     # Script-hardening
182:     set -o errexit
183:     set -o errtrace
184:     set -o nounset
185:     set -o pipefail
186:     # bash 5.0: trap exit: ${FUNCNAME[0]}: ongebonden variabele
187:     trap 'signal error $LINENO ${FUNCNAME:--} "$BASH_COMMAND" $?' ERR
188:     trap 'signal exit $LINENO ${FUNCNAME:--} "$BASH_COMMAND" $?' EXIT
189:     trap 'signal sighup $LINENO ${FUNCNAME:--} "$BASH_COMMAND" $?' SIGHUP # 1
190:     trap 'signal sigint $LINENO ${FUNCNAME:--} "$BASH_COMMAND" $?' SIGINT # 2
191:     trap 'signal sigterm $LINENO ${FUNCNAME:--} "$BASH_COMMAND" $?' SIGTERM #15
192:
193:     # Less-opties, overgenomen (en aangepast, zie 'man less', zoek PROMPTS)
194:     # van:
195:     # 1. systemctl en journalctl, zie bijv. 'man systemctl', zoek LESS
196:     # ("FRSXMK")
197:     # 2. man, zie 'man man', zoek LESS
198:     # shellcheck disable=SC2034
199:     LESS_OPTIONS="--LONG-PROMPT --no-init --quit-if-one-screen --quit-on-intr \
200: --RAW-CONTROL-CHARS --prompt=M Tekst uitvoer $PROGNAME \
201: ?ltregel %lt?L/%L:.byte %bB?s/%s..?e (EINDE) :?pB %pB%. .(druk h voor \
202: hulp of q voor stoppen)"
203:     # shellcheck disable=SC2034
204:     USAGELINE="Typ '$PROGNAME --usage' voor meer informatie."
```

```
205: }
206:
207: log() {
208:     printf '%b' "$@" |& $LOGCMD
209: }
210:
211: log_debug() {
212:     printf '%b' "$@" |& $LOGCMD_DEBUG
213: }
214:
215: process_general_options() {
216:     while true; do
217:         case $1 in
218:             -d|--debug)
219:                 process_option_debug
220:                 shift
221:                 ;;
222:             -g|--gui)
223:                 OPTION_GUI=true
224:                 shift
225:                 ;;
226:             -h|--help)
227:                 OPTION_HELP=true
228:                 shift
229:                 ;;
230:             -u|--usage)
231:                 OPTION_USAGE=true
232:                 shift
233:                 ;;
234:             -v|--version)
235:                 OPTION_VERSION=true
236:                 shift
237:                 ;;
238:             --)
239:                 break
240:                 ;;
241:             *)
242:                 shift
243:                 ;;
244:         esac
245:     done
246:
247:     if $OPTION_HELP; then
248:         process_option_help
249:         exit $SUCCESS
250:     elif $OPTION_USAGE; then
251:         process_option_usage
252:         exit $SUCCESS
253:     elif $OPTION_VERSION; then
254:         process_option_version
255:         exit $SUCCESS
256:     fi
257: }
258:
259: process_option_debug() {
260: # Enable code-stepping.
261: #     trap '(read -p "[${BASH_SOURCE:$LINENO}] $BASH_COMMAND?)"' DEBUG
262:
263:     printf "${YELLOW}%s\n${NORMAL}" '***** START DEBUG-SESSIE *****'
264:     log_debug 'START DEBUG-SESSIE'
265:     log_debug 'Start show current environment'
266:     log_debug 'uname --all:'
267:     log_debug "$(uname --all)"
268:     log_debug 'lsb_release --all:'
269:     log_debug "$(lsb_release --all 2> >($LOGCMD_DEBUG))"
270:     log_debug 'cat /etc/os-release:'
271:     log_debug "$(cat /etc/os-release)"
272:     log_debug 'declare -p:'
```

```
273:     log_debug    "$ (declare -p) "
274:     log_debug    'End show current environment'
275:     log_debug    'Routing bash xtrace to FD4'
276:     exec 4>      >($LOGCMD_DEBUG)
277:     BASH_XTRACEFD=4
278:     log          'Setting bash options'
279:     set          -o verbose
280:     set          -o xtrace
281:     log_debug    'Let the fun begin ...'
282:     # shellcheck disable=SC2016
283:     ps4='+ (${BASH_SOURCE}:${LINENO}): ${FUNCNAME[0]:+${FUNCNAME[0]}}(): {'
284:     export PS4=$ps4
285:     unset ps4
286:
287:     OPTION_DEBUG=true
288: }
289:
290: process_option_help() {
291:     printf "%s\n\n%s\n" "$HELP" "Typ 'man $PROGNAME' voor meer informatie."
292: }
293:
294: process_option_usage() {
295:     printf "%s\n\n%s\n" "$USAGE" "Typ '$PROGNAME --help' voor meer informatie."
296: }
297:
298: process_option_version() {
299:     local release_number='00.00.00'
300:     local copyright_years='1970'
301:     local release_date='1970-01-01'
302:
303:     copyright_years=$(
304:         grep    --regex='Auteursrecht '    \
305:             "$PROGDIR/$PROGNAME"          |
306:         awk '{print $4;exit}'
307:     )
308:     release_number=$(
309:         awk -FRelNum=    \
310:             '/RelNum=/{print $2;exit}'    \
311:             "$PROGDIR/$PROGNAME"
312:     )
313:     release_number=${release_number:0:8}
314:     release_date=$(
315:         awk -FRelDat=    \
316:             '/RelDat=/{print $2;exit}'    \
317:             "$PROGDIR/$PROGNAME"
318:     )
319:     release_date=${release_date:0:10}
320:     info "$PROGNAME $release_number ($release_date)
321:
322: Geschreven door Karel Zimmer <info@karelzimmer.nl>.
323:
324: Auteursrecht (c) $copyright_years Karel Zimmer.
325: GNU Algemene Publieke Licentie <https://www.gnu.org/licenses/gpl.html>."
326: }
327:
328: signal() {
329:     local signal=${1:-signal?}
330:     local -i lineno=${2:-0}
331:     local function=${3:-funcname?}
332:     local command=${4:-command?}
333:     local -i rc=${5:-1}
334:     local apt_repair=''
335:     local rc_desc=''
336:     local -i rc_desc_signalno=0
337:     local status="${RED}$ERROR/ERROR${NORMAL}"
338:
339:     case $rc in
340:         0)
```

```
341:         rc_desc='successful termination'
342:         status="${GREEN}$rc/SUCCESS${NORMAL}"
343:         ;;
344:     1)
345:         rc_desc='terminated with error'
346:         ;;
347:     2)
348:         rc_desc='terminated with warning'
349:         status="${YELLOW}$rc/WARNING${NORMAL}"
350:         ;;
351:     6[4-9]|7[0-8])                # 64 - 78
352:         rc_desc="open file '/usr/include/sysexits.h' and look for '$rc'"
353:         ;;
354:     126)
355:         rc_desc='command cannot execute'
356:         ;;
357:     127)
358:         rc_desc='command not found'
359:         ;;
360:     128)
361:         rc_desc='invalid argument to exit'
362:         ;;
363:     129)                # sighup (128+ 1)
364:         rc_desc='hangup'
365:         ;;
366:     130)                # sigint (128+ 2)
367:         rc_desc='terminated by control-c'
368:         ;;
369:     13[1-9]|14[0-2])    # 131 (128+3) - 142 (128+14)
370:         rc_desc_signalno=$((rc - 128))
371:         rc_desc="typ 'trap -1' and look for '$rc_desc_signalno'"
372:         ;;
373:     143)                # sigterm (128+15)
374:         rc_desc='termination signal'
375:         ;;
376:     14[4-9]|1[5-8][0-9]|19[0-2]) # 144 (128+16) - 192 (128+64)
377:         rc_desc_signalno=$((rc - 128))
378:         rc_desc="typ 'trap -1' and look for '$rc_desc_signalno'"
379:         ;;
380:     255)
381:         rc_desc='exit status out of range'
382:         ;;
383:     *)
384:         rc_desc='unknown error'
385:         ;;
386: esac
387:
388:     log_debug "signal=$signal, line=$lineno, function=$function, \
389: command=$command, code=$rc/$rc_desc$aapt_repair"
390:
391:     case $signal in
392:         error)
393:             exit "$rc"
394:             ;;
395:         exit)
396:             signal_exit_cleanup
397:             if $OPTION_DEBUG; then
398:                 set          +o verbose
399:                 set          +o xtrace
400:                 BASH_XTRACEFD=''
401:                 exec          4>&-
402:                 printf       "${YELLOW}%s\n${NORMAL}" \
403:                     '***** EINDE DEBUG-SESSIE *****'
404:                 printf       '%s\n' "Opdracht om de log uit te lezen:
405: ${BLUE}$CHKLOGCMD${NORMAL}"
406:                 log          'EINDE DEBUG-SESSIE'
407:             fi
408:             log "ended (code=exited, status=$status)"
```

```
409:
410:     # Een non-gui script gestart met optie gui.
411:     if $TERMINAL; then
412:         read -rp "Druk op de Enter-toets om dit scherm te sluiten \
413: [Enter]: " < /dev/tty
414:     fi
415:
416:     trap - ERR EXIT SIGINT SIGPIPE SIGTERM SIGHUP
417:     exit "$rc"
418:     ;;
419: sig*)
420:     exit "$rc"
421:     ;;
422: esac
423: }
424:
425: signal_exit_cleanup() {
426:     case $PROGNAME in
427:         kzbackup)
428:             if [[ -e $BACKUP_TO_DELETE ]]; then
429:                 info "Programma kzbackup is onderbroken.
430: Opgeslagen backup wordt verwijderd."
431:                 rm "$BACKUP_TO_DELETE" |& $LOGCMD
432:                 info 'Opgeslagen backup is verwijderd.'
433:             fi
434:             ;;
435:         kzinstall)
436:             apt_repair="Als de pakketbeheerder 'apt' foutmeldingen geeft,
437: start dan een Terminalvenster, en voer uit:
438:     ${BLUE}sudo apt-get update --fix-missing${NORMAL}
439:     ${BLUE}sudo dpkg --configure --pending${NORMAL}
440:     ${BLUE}sudo apt-get install --fix-broken${NORMAL}
441:     ${BLUE}sudo update-initramfs -u${NORMAL}"
442:             printf '%s' "${NORMAL}${CURSOR_VISIBLE}"
443:             ;;
444:         kzsetup)
445:             printf '%s' "${NORMAL}${CURSOR_VISIBLE}"
446:             ;;
447:     esac
448: }
449:
450: warning() {
451:     local text=${1:-text?}
452:
453:     if $OPTION_GUI; then
454:         # Constructie '2> >($LOGCMD)' om stderr naar de log te krijgen.
455:         # Voorbeeld: Unable to init server: Kon niet verbinden:
456:         #     Verbinding is geweigerd
457:         #     en: (zenity:47712): Gtk-WARNING **: 10:35:49.339:
458:         #     cannot open display:
459:         if ! zenity --warning \
460:             --no-markup \
461:             --width 500 \
462:             --height 100 \
463:             --title "Waarschuwing $PROGNAME" \
464:             --text "$text" \
465:             --ok-label 'OkÃ©' \
466:             2> >($LOGCMD); then
467:             true
468:         fi
469:     else
470:         printf "${YELLOW}%b${NORMAL}\n" "$text" >&2
471:     fi
472: }
473:
474: # EOF
```