

```

# shellcheck shell=bash
#####
# Algemene module voor shell scripts.
#
# Geschreven in 2009 door Karel Zimmer <info@karelzimmer.nl>, Creative Commons
# Publiek Domein Verklaring <http://creativecommons.org/publicdomain/zero/1.0>.
#####

#####
# Constants
#####

declare module_name='kz_common.sh'
declare module_desc='Algemene module voor shell scripts'

declare -i ok=0
declare -i err=1

#####
# Variables
#####

declare options_short='huv'
declare options_long='help,usage,version'
declare options_usage='[-h|--help] [-u|--usage] [-v|--version]'
declare options_help="  -h, --help      toon deze hulptekst
  -u, --usage    toon een korte gebruikssamenvatting
  -v, --version  toon de programmaversie"

declare -a cmdline_args=()
declare less_options=''
declare logcmd_check=''
declare logcmd=''
declare option_gui=false
# pkexec needs absolute path-name, e.g. ./script -> /path/to/script.
declare program_exec=${0/#./$program_path}
declare text=''
declare title=''
declare usage_line=''

# Terminalattributen, zie man terminfo. Gebruik ${<variabele-naam>}.
declare blink=''
declare blue=''
declare cursor_invisable=''
declare cursor_visable=''
declare green=''
declare normal=''
declare red=''
declare rewrite_line=''
declare yellow=''

#####
# Functions
#####

function kz_common.check_dpkgd_snapd {
    local -i dpkg_wait=10

    if find /snap/core/*/var/cache/debconf/config.dat &> /dev/null; then
        # Systeem met snaps.
        while sudo fuser

```

```

        /var/{lib/{dpkg,apt/lists},cache/apt/archives}/lock \
        /var/cache/debconf/config.dat \
        /snap/core/*/var/cache/debconf/config.dat \
        &> /dev/null; do
    log "Wacht ${dpkg_wait}s tot andere pakketbeheerder klaar is..."
    sleep $dpkg_wait
done
else
# Stelsysteem zonder snaps.
while sudo fuser \
    /var/{lib/{dpkg,apt/lists},cache/apt/archives}/lock \
    /var/cache/debconf/config.dat \
    &> /dev/null; do
    log "Wacht ${dpkg_wait}s tot andere pakketbeheerder klaar is..."
    sleep $dpkg_wait
done
fi
}

```

```

function kz_common.check_on_ac_power {
    local -i on_battery=0

    on_ac_power |& $logcmd || on_battery=$?
    if [[ on_battery -eq 1 ]]; then
        warn '
De computer gebruikt nu alleen de accu voor de stroomvoorziening.
Geadviseerd wordt om de computer aan te sluiten op het stopcontact.'
        kz_common.wait_for_enter
    fi
}

```

```

function kz_common.wait_for_enter {
    if $option_gui; then
        return
    fi
    read -rp '
Druk op de Enter-toets om door te gaan [Enter]: ' < /dev/tty
}

```

```

function kz_common.check_user_root {
    local -i pkexec_rc=0

    if ! kz_common.check_user_sudo; then
        info 'Reeds uitgevoerd door de beheerder.'
        exit $ok
    fi
    if [[ $UID -ne 0 ]]; then
        if $option_gui; then
            log "restarted (pkexec $program_exec ${cmdline_args[*]})" \
                --priority=debug
            pkexec "$program_exec" "${cmdline_args[@]}" || pkexec_rc=$?
            exit $pkexec_rc
        else
            log "restarted (exec sudo $program_exec ${cmdline_args[*]})" \
                --priority=debug
            if ! sudo -n true &> /dev/null; then
                printf '%s\n' "Authenticatie is vereist om \
${display_name:-$module_name} uit te voeren."
            fi
        fi
    fi
}

```

```

        exec sudo "$program_exec" "${cmdline_args[@]}"
    fi
fi
}

function kz_common.check_user_sudo {
    # Mag gebruiker sudo uitvoeren?
    if [[ $UID -eq 0 ]]; then
        # Voor de "grace"-periode van sudo, of als root.
        return $ok
    elif groups "$USER" | grep --quiet --regexp='sudo'; then
        return $ok
    else
        return $err
    fi
}

function kz_common.developer {
    local action=${1:-check}

    if [[ $action = 'check' ]]; then
        # Aangemeld als ontwikkelaar?
        if [[ $USER = 'karel' && $HOSTNAME == pc?? ]]; then
            return $ok
        else
            return $err
        fi
    else
        printf '%s\n' "Alleen uitvoeren als Ontwikkelaar, d.i. aangemeld als \
karel op pcNN (bijvoorbeeld pc01).\"
    fi
}

function kz_common.init_script {
    # Script-hardening.
    set -o errexit
    set -o errtrace
    set -o nounset
    set -o pipefail

    logcmd="systemd-cat --identifier=${program_name:-$module_name}"
    logcmd_check="journalctl --all --boot \
--identifier=${program_name:-$module_name} \
--since='$(date '+%Y-%m-%d %H:%M:%S')'"

    trap 'signal err      $LINENO ${FUNCNAME:--} "$BASH_COMMAND" $?' err
    trap 'signal exit     $LINENO ${FUNCNAME:--} "$BASH_COMMAND" $?' EXIT
    trap 'signal sighup   $LINENO ${FUNCNAME:--} "$BASH_COMMAND" $?' SIGHUP # 1
    trap 'signal sigint   $LINENO ${FUNCNAME:--} "$BASH_COMMAND" $?' SIGINT # 2
    trap 'signal sigpipe  $LINENO ${FUNCNAME:--} "$BASH_COMMAND" $?' SIGPIPE #13
    trap 'signal sigterm  $LINENO ${FUNCNAME:--} "$BASH_COMMAND" $?' SIGTERM #15

    log "started ($program_exec $* as $USER)" --priority=notice

    if [[ $(lsb_release --id --short) = 'Debian' && $UID -ne 0 ]]; then
        xhost +si:localuser:root |& $logcmd
    fi

    if [[ -t 1 ]]; then
        set_terminal_attributes
    fi
}

```

```

cmdline_args=("$@")
less_options="--LONG-PROMPT --no-init --quit-if-one-screen --quit-on-intr \
--RAW-CONTROL-CHARS --prompt=MTekstuitvoer ${display_name:-$module_name} \
?ltregel %lt?L van %L.:byte %bB?s van %s..? .?e (EINDE) :?pB %pB\%. .(druk op \
h voor hulp of q om te stoppen)"
usage_line="Typ '${display_name:-$module_name}' --usage' voor meer \
informatie."
}

```

```

function signal {
    local signal=${1:-unknown}
    local -i lineno=${2:-unknown}
    local function=${3:-unknown}
    local command=${4:-unknown}
    local -i rc=${5:-$err}
    local rc_desc=''
    local -i rc_desc_signalno=0
    local status="${red}$rc/err${normal}"

case $rc in
    0)
        rc_desc='successful termination'
        status="${green}$rc/ok${normal}"
        ;;
    1)
        rc_desc='terminated with error'
        ;;
    6[4-9]|7[0-8])
        # 64--78
        rc_desc="open file '/usr/include/sys/exits.h' and look for '$rc'"
        ;;
    126)
        rc_desc='command cannot execute'
        ;;
    127)
        rc_desc='command not found'
        ;;
    128)
        rc_desc='invalid argument to exit'
        ;;
    129)
        # SIGHUP (128+1)
        rc_desc='hangup'
        ;;
    130)
        # SIGINT (128+2)
        rc_desc='terminated by control-c'
        ;;
    13[1-9]|140)
        # 140 (128+12)
        rc_desc_signalno=$((rc - 128))
        rc_desc="typ 'trap -l' and look for $rc_desc_signalno"
        ;;
    141)
        # SIGPIPE (128+13)
        rc_desc='broken pipe: write to pipe with no readers'
        ;;
    142)
        # SIGALRM (128+14)
        rc_desc='timer signal from alarm'
        ;;
    143)
        # SIGTERM (128+15)
        rc_desc='termination signal'
        ;;
    14[4-9]|1[5-8][0-9]|19[0-2])
        # 144 (128+16)--192 (128+64)
        rc_desc_signalno=$((rc - 128))
        rc_desc="typ 'trap -l' and look for $rc_desc_signalno"
        ;;

```

```

255)
    rc_desc='exit status out of range'
    ;;
*)
    rc_desc='unknown error'
    ;;
esac
log "signal: $signal, line: $lineno, function: $function, command: \
$command, code: $rc ($rc_desc)" --priority=debug

case $signal in
err)
    err "
In programma $program_name is een fout opgetreden."
    exit "$rc"
    ;;
exit)
    signal_exit
    log "ended (code=exited, status=$status)" --priority=notice
    trap - err EXIT SIGHUP SIGINT SIGPIPE SIGTERM
    exit "$rc"
    ;;
*)
    err "
Programma $program_name is onderbroken."
    exit "$rc"
    ;;
esac
}

```

```

function signal_exit {
    local apt_err="Als de pakketbeheerder apt foutmeldingen geeft, start een \
Terminalvenster en voer uit:
[1] ${blue}kz update${normal}
[2] ${blue}sudo update-initramfs -u${normal}"

    case $program_name in
kz-getdeb)
        # Verwijder niet kz en kz.1 i.v.m. script kz en man-pagina kz.1.
        rm --force kz.{2..99} /tmp/kz_common.sh
        # Maar wel als in HOME, zoals beschreven in Checklist installatie.
        cd "$HOME"
        rm --force kz kz.1

        if [[ $rc -ne $ok ]]; then
            log "$apt_err" --priority=debug
        fi
        ;;
kz-install)
        printf "${normal}%s" "${cursor_visable}"

        if [[ $rc -ne $ok ]]; then
            log "$apt_err" --priority=debug
        fi
        ;;
kz-setup)
        printf "${normal}%s" "${cursor_visable}"
        ;;
esac
}

```

```

function set_terminal_attributes {

```

```

blink=$(tput bold; tput blink)
normal=$(tput sgr0)
blue=$(tput bold; tput setaf 4)
cursor_invisible=$(tput civis)
cursor_visible=$(tput cvvis)
green=$(tput bold; tput setaf 2)
red=$(tput bold; tput setaf 1)
rewrite_line=$(tput cuu1; tput el)
yellow=$(tput bold; tput setaf 3)
}

```

```

function kz_common.process_options {
    while true; do
        case $1 in
            -h|--help)
                kz_common.process_option_help
                exit $ok
                ;;
            -u|--usage)
                kz_common.process_option_usage
                exit $ok
                ;;
            -v|--version)
                kz_common.process_option_version
                exit $ok
                ;;
            --)
                break
                ;;
            *)
                shift
                ;;
        esac
    done
}

```

```

function kz_common.process_option_help {
    info "${help:-Variable help not set}"
}

```

```

Typ 'man ${display_name:-$module_name}' voor meer informatie."
}

```

```

function kz_common.process_option_usage {
    info "${usage:-Variable usage not set}"
}

```

```

Typ '${display_name:-$module_name} --help' voor meer informatie."
}

```

```

function kz_common.process_option_version {
    local build_id='1970-01-01'
    local program_year='1970'

    if [[ -e /usr/local/etc/kz-build-id ]]; then
        build_id=$(cat /usr/local/etc/kz-build-id)
    fi
    program_year=$(
        grep '# Geschreven in ' "$program_path/$program_name" |
        cut --delimiter=' ' --fields=4
    ) || true
    if [[ $program_year = '' ]]; then

```

```

        program_year='1970'
    fi
    info "$program_name (kz) 365 ($build_id)

Geschreven in ${program_year:-1970} door Karel Zimmer <info@karelzimmer.nl>, \
Creative Commons
Publiek Domein Verklaring <http://creativecommons.org/publicdomain/zero/1.0>."
}

```

```

function kz_common.reset_terminal_attributes {
    blink=''
    blue=''
    cursor_invisable=''
    cursor_visable=''
    green=''
    normal=''
    red=''
    rewrite_line=''
    yellow=''
}

```

```

function log {
    printf '%b\n' "$1" |& $logcmd
}

```

```

function info {
    local title="Informatie ${display_name:-$module_name}"

    if $option_gui; then
        zenity --info \
            --no-markup \
            --width 600 \
            --height 100 \
            --title "$title" \
            --text "$@" \
            --ok-label 'Oké' 2> >($logcmd) || true
    else
        printf '%b\n' "$@"
    fi
    log "$@" --priority=info
}

```

```

function warn {
    local title="Waarschuwing ${display_name:-$module_name}"

    if $option_gui; then
        zenity --warning \
            --no-markup \
            --width 600 \
            --height 100 \
            --title "$title" \
            --text "$@" \
            --ok-label 'Oké' 2> >($logcmd) || true
    else
        printf "${yellow}%b\n${normal}" "$@" >&2
    fi
    log "$@" --priority=warn
}

```

```

function err {
    local title="Foutmelding ${display_name:-$module_name}"

    if $option_gui; then
        zenity --error \
              --no-markup \
              --width 600 \
              --height 100 \
              --title "$title" \
              --text "$@" \
              --ok-label 'Oké' 2> >($logcmd) || true
    else
        printf "${red}%b\n${normal}" "$@" >&2
    fi
    log "$@" --priority=err
}

{
    # Anonymous function to avoid using shellcheck directive disable=SC2034.
    echo "$blink"
    echo "$cursor_invisable"
    echo "$less_options"
    echo "$logcmd_check"
    echo "$module_desc"
    echo "$options_help"
    echo "$options_long"
    echo "$options_short"
    echo "$options_usage"
    echo "$rewrite_line"
    echo "$text"
    echo "$usage_line"
} > /dev/null

true

```