

```

#!/bin/bash
# shellcheck source=kz_common.sh
#####
# Execute IVP.
#
# Written by Karel Zimmer <info@karelzimmer.nl>, CC0 1.0 Universal
# <https://creativecommons.org/publicdomain/zero/1.0>, 2023.
#####
#####
# Import
#####

program_path=$(cd "$(dirname "$(realpath "$0")")" && pwd)
source "$program_path"/kz_common.sh 2> >(systemd-cat) || exit 1

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

program_name='kz-ivp'
program_desc=$(gettext 'Execute IVP')
display_name=${program_name/kz-/kz }

usage=$(eval_gettext "Usage: \${display_name} \${options_usage}")
help="$(eval_gettext "Usage: \${display_name} [OPTION...])"

$program_desc.

$(gettext 'Options:')
$options_help"

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

function check_input {
    local -i rc=0
    local parsed=''

    parsed=$(
        getopt --alternative          \
              --options               "$options_short"  \
              --longoptions           "$options_long"   \
              --name                   "$display_name"  \
              --                       "$@"
    ) || rc=$?
    if [[ $rc -ne $ok ]]; then
        printf '%s\n' "$usage_line"
        exit $error
    fi
    eval set -- "$parsed"
    process_option "$@"

    while true; do
        case $1 in
            --)
                shift
                break
            ;;
            *)
                shift
        esac
    done
}

```

```

        ;;
    esac
done

if [[ -n "$*" ]]; then
    printf "$display_name: $*: %s\n$usage_line\n" \
        "$$(gettext 'arguments are not allowed')"
    exit $error
fi
}

function process_input {
    local dir=''
    local scriptname=''

    # Message catalogue (gettext).
    dir=/usr/share/locale/nl/LC_MESSAGES
    if ! [[ -e $dir/kz.mo ]]; then
        error "$$(eval_gettext "Message catalog \ $dir/kz.mo not found.")"
    fi

    # Check all kz scripts.
    while read -r script; do
        scriptname=$(basename "$script")
        check_script
    done <<(
        find /usr/bin \
            -name 'kz*' \
            -type f \
            -executable \
            -print \
            |
        sort
    )
}

function check_script {
    local gui='-''-'G''U''I'
    local root=' kz_common.''c''h''e''c''k''_'u''s''e''r''_'r''o''o''t'

    # Script.
    if ! "$scriptname" --version |& $logcmd; then
        error "$$(eval_gettext "Script \ $scriptname returned an error.")"
        maxrc=$error
    fi

    # Desktop file.
    if grep --quiet --word-regexp --regexp="$gui" "/usr/bin/$scriptname"; then
        dir=/usr/share/applications
        if ! [[ -e $dir/$scriptname.desktop ]]; then
            error "$$(eval_gettext "Desktop configuration file \
            \ $dir/\ $scriptname.desktop not found.")"
            maxrc=$error
        fi
    fi

    # PolicyKit action definition file.
    if grep --quiet --word-regexp --regexp="$root" "/usr/bin/$scriptname"; then
        dir=/usr/share/polkit-1/actions
        if ! [[ -e $dir/nl.karelzimmer.$scriptname.policy ]]; then
            error "$$(eval_gettext "PolicyKit action definition file \
            \ $dir/nl.karelzimmer.\ $scriptname.policy not found.")"
            maxrc=$error
        fi
    fi
}

```

```

        fi
    fi

    # Tab-completion script.
    dir=/usr/share/bash-completion/completions
    if ! [[ -e $dir/$scriptname ]]; then
        error "$(eval_gettext "Tab completion script \${dir}/\${scriptname} not \
found.")"
        maxrc=$error
    fi

    # Manual page (EN + NL).
    dir=/usr/share/man/man1
    if ! [[ -e $dir/$scriptname.1.gz ]]; then
        error "$(eval_gettext "Manual \${dir}/\${scriptname}.1.gz not found.")"
        maxrc=$error
    fi
    dir=/usr/share/man/nl/man1
    if ! [[ -e $dir/$scriptname.1.gz ]]; then
        error "$(eval_gettext "Manual \${dir}/\${scriptname}.1.gz not found.")"
        maxrc=$error
    fi
}

function term_script {
    exit $maxrc
}

#####
# Main
#####

function main {
    init_script "$@"
    check_input "$@"
    process_input
    term_script
}

main "$@"

```