

```

#!/bin/bash
# shellcheck source=kz_common.sh
#####
# Run command.
#
# Written by Karel Zimmer <info@karelzimmer.nl>, CC0 1.0 Universal
# <https://creativecommons.org/publicdomain/zero/1.0>, 2021-2023.
#####
#####
# Import
#####

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

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

program_name='kz'
program_desc=$(gettext 'Run command')
display_name=${program_name/kz-/kz }

usage=$(eval_gettext "Usage: \${display_name} COMMAND [ARGUMENT...]
or: \${display_name} [-l|--list] [-u|--usage] [-h|--help] [-v|--version]")

help=$(eval_gettext "Usage: \${display_name} COMMAND [ARGUMENT...]
or: \${display_name} OPTION")

$program_desc.

$(gettext "Options:
-l, --list      show available COMMANDs")
$options_help

$(gettext "Arguments:
COMMAND        execute COMMAND
ARGUMENT       ARGUMENTs for COMMAND")

exec_program_name=' '

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

function check_input {
    # No getopt here like in all other scripts; any specified options in
    # ARGUMENT, with kz COMMAND ARGUMENT, are for COMMAND, not for kz.
    if [[ -z "$*" ]]; then
        printf "%s\n" "$usage"
        exit $error
    fi
    case $1 in
        -h*|--h*)
            process_option_help
            exit $ok
            ;;
        -u*|--u*)
            process_option_usage
            exit $ok
            ;;
        -v*|--v*)
    
```

```

        process_option_version
        exit $ok
        ;;
    -l*|--l*)
        process_option_list
        exit $ok
        ;;
    -*)
        printf "$display_name: %s: $1%s\n$usage_line\n" \
            "$(gettext 'invalid option')"
        exit $error
        ;;
    *)
        exec_program_name=$1
        ;;
esac
if ! [[ -f "$program_path/kz-$exec_program_name" ]]; then
    printf "$display_name: $exec_program_name: %s\n$usage_line\n" \
        "$(gettext 'command does not exist')"
    exit $error
fi
}

```

```

function process_option_list {
    info "$(gettext 'The following COMMANDs are available:

    COMMAND')"
    find "$program_path"/kz-* \
        -maxdepth 1 \
        -type f \
        -executable \
        -readable \
        -not -name '*~' \
        -not -name 'kz' \
        -printf '%f\n' \
    sort \
    sed --expression='s/kz-//' \
        nl --number-width=2 \
            --number-format=rn \
            --number-separator=] ' \
            --body-numbering=a \
        sed --expression='s/^/[/'
    info "
$(gettext 'To execute a COMMAND:') ${blue}$display_name COMMAND${normal}"
}

```

```

function process_input {
    shift
    exec "$program_path/kz-$exec_program_name" "$@"
}

```

```

function term_script {
    exit $ok
}

```

```

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

function main {

```

```
    init_script "$@"  
    check_input "$@"  
    process_input "$@"  
    term_script  
}  
main "$@"
```