

```

#!/bin/bash
# shellcheck shell=bash
#####
# Install kz package.
#
# Written in 2016 by Karel Zimmer <info@karelzimmer.nl>, Creative Commons
# Public Domain Dedication <https://creativecommons.org/publicdomain/zero/1.0>.
#####

export TEXTDOMAIN=kz TEXTDOMAINDIR=/usr/share/locale
# shellcheck source=/dev/null
source /usr/bin/gettext.sh

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

declare program_name='kz-getdeb'
declare program_desc
program_desc=$(gettext 'Install kz package')
declare display_name=${program_name/kz-/kz }

declare bold
bold=$(tput bold)
declare normal
normal=$(tput sgr0)

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

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

function init_script {
    set -o errexit
    set -o errtrace
    set -o nounset
    set -o pipefail
}

function check_input {
    if [[ "$*" ]]; then
        printf "$display_name: $*: %s\n%s\n" \
            "$(gettext 'arguments are not allowed')" \
            "$(eval_gettext "Usage: \${display_name} \${options_usage}")"
        # For kz ivp, exit 0 instead of exit 1.
        exit 0
    fi
}

function process_input {
    if groups "$USER" | grep --quiet --regexp='sudo'; then
        sudo true
    else
        printf '%s\n' "$(gettext 'Already performed by the administrator.')"
        term_script
    fi
}

```

```

install_package
}

function install_package {
    local site_deb='https://karelzimmer.nl/downloads/kz/kz_365_all.deb'
    local temp_deb=''

    temp_deb=$(mktemp -t "$program_name-XXXXXXXXXX.deb")
    # Prevent "N: ... user '_apt'. - pkgAcquire::Run (13: Permission denied)".
    chmod o+r "$temp_deb"
    printf '%s\n' "*" ${bold}Download${normal} ($display_name)"
    wget --output-document="$temp_deb" $site_deb
    check_dpkgd_snapd
    printf '%s\n' "*" ${bold}Install${normal} ($display_name)"
    sudo DEBIAN_FRONTEND=noninteractive \
        apt-get \
        reinstall \
        --yes \
        "$temp_deb"
    rm "$temp_deb"
}

```

```

function check_dpkgd_snapd {
    local -i dpkg_wait=10
    local text
    text=$(eval_gettext \
        "Wait \${dpkg_wait}s for another package manager to finish...")

    if find /snap/core/*/var/cache/debconf/config.dat &> /dev/null; then
        # System with 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
            printf '%s\n' "$text"
            sleep $dpkg_wait
        done
    else
        # System without snaps.
        while sudo fuser \
            /var/{lib/{dpkg,apt/lists},cache/apt/archives}/lock \
            /var/cache/debconf/config.dat \
            &> /dev/null; do
            printf '%s\n' "$text"
            sleep $dpkg_wait
        done
    fi
}

```

```

function term_script {
    local title
    title=$(gettext 'Finished')
    local text
    text=$(gettext "Package kz has been successfully installed.")
}

```

Now follow the steps as described
in the Checklist installation.

Checklist installation can be found
on the site <https://karelzimmer.nl>,

under Linux.

Type 'exit' to close this window.")

```
clear -x
TERM=ansi whiptail --backtitle "$display_name - $program_desc" \
                  --title      "$title" \
                  --infobox    "$text" \
                  16 47

# Do not delete kz and kz.1 due to scripts and manuals with the same name.
rm --force kz.{2..99}
# But do delete them if in HOME, as described in Checklist installation.
rm --force "$HOME"/kz "$HOME"kz.1

exit 0
}

#####
# Script
#####

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

main "$@"
```