

```
1: #!/bin/bash
2: # shellcheck source=kzcommon.sh
3: # -----
4: # Applijst maken.
5: #
6: # Geschreven door Karel Zimmer <info@karelzimmer.nl>.
7: #
8: # Auteursrecht (c) 2013-2021 Karel Zimmer.
9: # GNU Algemene Publieke Licentie <https://www.gnu.org/licenses/gpl.html>.
10: #
11: # RelNum=18.04.00
12: # RelDat=2021-01-18
13: # -----
14: #
15: # -----
16: # Global constants
17: # -----
18: source "$(dirname "$0")"/kzcommon.sh
19: readonly OUTPUTFILE_1=/root/$PROGNAME-$HOSTNAME-deb-bestanden
20: readonly OUTPUTFILE_2=/root/$PROGNAME-$HOSTNAME-pakketbronnen
21: readonly OUTPUTFILE_3=/root/$PROGNAME-$HOSTNAME-pakketten
22: readonly OUTPUTFILE_4=/root/$PROGNAME-$HOSTNAME-favorieten
23: readonly OUTPUTFILE_5=/root/$PROGNAME-$HOSTNAME-snaps
24: readonly RUN_AS_SUPERUSER=true
25:
26: # Bij aanpassingen ook .completion aanpassen!
27: readonly OPTIONS_SHORT=$OPTIONS_SHORT_COMMON
28: readonly OPTIONS_LONG=$OPTIONS_LONG_COMMON
29: readonly USAGE="Gebruik: $PROGNAME $OPTIONS_USAGE_COMMON"
30: readonly HELP="Gebruik: $PROGNAME [OPTIE...]"
31:
32: Applijst maken.
33:
34: Opties:
35: $OPTIONS_HELP_COMMON"
36:
37: # -----
38: # Global variables
39: # -----
40:
41: # -----
42: # Functions
43: # -----
44: check_input() {
45:     local -i getopt_rc=0
46:     local parsed=''
47:
48:     parsed=$(
49:         getopt --alternative           \
50:                --options               "$OPTIONS_SHORT"      \
51:                --longoptions           "$OPTIONS_LONG"       \
52:                --name                  "$PROGNAME"           \
53:                -- "$@"
54:     ) || getopt_rc=$?
55:     if [[ $getopt_rc -ne 0 ]]; then
56:         printf '%s\n' "$USAGELINE" >&2
57:         exit $ERROR
58:     fi
59:     eval set -- "$parsed"
60:     process_general_options "$@"
61:
62:     while true; do
63:         case $1 in
64:             --)
65:                 shift
66:                 break
67:             ;;
68:             *)
```

```
69:             shift
70:             ;;
71:     esac
72: done
73:
74: if [[ "$*" ]]; then
75:     printf "$PROGNAME: %s\n%s\n" 'geen argumenten opgeven' "$USAGELINE" >&2
76:     exit $ERROR
77: fi
78:
79: check_user
80: # Na check_user voor toegang tot media.
81: remove_old_configuration
82: }
83:
84: remove_old_configuration() {
85:     rm --force "$HOME"/.lsinstal*"$HOSTNAME"* /root/.lsinstal*-"$HOSTNAME"*
86: }
87:
88: process_input() {
89:     local title='Applijst maken'
90:     local text0='Opdrachten worden verwerkt'
91:     local text1="[1/5] Bepaal welke deb-bestanden (Debian-pakketten) zijn \
92: geÃnstalleerd"
93:     local text2="[2/5] Bepaal welke pakketbronnen (\\"repositories\\" of \
94: \\"repo's\\") zijn toegevoegd"
95:     local text3="[3/5] Bepaal welke pakketten door de gebruiker zijn \
96: geÃnstalleerd"
97:     local text4=' [4/5] Bepaal welke favorieten in gebruik zijn'
98:     local text5=' [5/5] Bepaal welke snaps zijn geÃnstalleerd'
99:
100: # Constructie '2> >($LOGCMD)' om stderr naar de log te krijgen.
101: # Voorbeeld: Unable to init server: Kon niet verbinden:
102: #           Verbinding is geweigerd
103: #           en: (zenity:47712): Gtk-WARNING **: 10:35:49.339:
104: #           cannot open display:
105: if $OPTION_GUI; then
106:     (
107:         printf '%s\n' "$text1"
108:         create_file_installed_without_repo
109:         printf '%s\n' "$text2"
110:         create_file_added_repos
111:         printf '%s\n' "$text3"
112:         create_file_installed_by_user
113:         printf '%s\n' "$text4"
114:         create_file_favorites
115:         printf '%s\n' "$text5"
116:         create_file_snaps
117:     ) |&
118:     zenity --progress \
119:           --pulsate \
120:           --auto-close \
121:           --no-cancel \
122:           --width 600 \
123:           --height 50 \
124:           --title "$title" \
125:           --text "$text0" \
126:           2> >($LOGCMD)
127: else
128:     create_file_installed_without_repo #1
129:     create_file_added_repos #2
130:     create_file_installed_by_user #3
131:     create_file_favorites #4
132:     create_file_snaps #5
133: fi
134: }
135:
136: create_file_installed_without_repo() {
```

```
137:     local number_of_lines=0
138:     local header="Dit zijn gedownloade deb-bestanden geïnstalleerd via \
139: Software, of met een opdracht zoals 'sudo apt-get --install <bestand.deb>'."
140:     local footer=''
141:
142:     printf '%s\n' "$text1 ..."
143:
144:     touch "$OUTPUTFILE_1" |& $LOGCMD
145:
146:     printf '%b\n' "$header\n$DASHES" > "$OUTPUTFILE_1"
147:
148:     apt-show-versions --initialize |& $LOGCMD
149:     if !     apt-show-versions           \
150:         --brief                               |
151:         grep      --word-regexp           \
152:                 --regexp='No available version' |
153:         cut      --delimiter=':'         \
154:                 --fields=1              >> "$OUTPUTFILE_1"; then
155:         true
156:     fi
157:
158:     number_of_lines=$(wc --lines < "$OUTPUTFILE_1")
159:     number_of_lines=$((number_of_lines - 2))
160:     footer="Dit bestand bevat $number_of_lines door de gebruiker \
161: geïnstalleerde deb-bestanden.
162: Meer via de opdracht: apt-cache show <bestand> | grep --regexp='Description:'"
163:     printf '%b\n' "$DASHES\n$footer" >> "$OUTPUTFILE_1"
164: }
165:
166: create_file_added_repos() {
167:     local number_of_lines=0
168:     local header="Deze pakketbronnen zijn meestal toegevoegd met de opdracht \
169: 'sudo add-apt-repository ppa:<ppa-gebruiker>/<ppa-naam>'."
170:     local footer=''
171:
172:     printf '%s\n' "$text2 ..."
173:
174:     touch "$OUTPUTFILE_2" |& $LOGCMD
175:
176:     printf '%b\n' "$header\n$DASHES" > "$OUTPUTFILE_2"
177:
178:     # shellcheck disable=SC2022,SC2062
179:     if !     grep      --recursive           \
180:         --no-filename           \
181:         --regexp='^deb '         \
182:         /etc/apt/sources.list*   |
183:         grep      --invert-match           \
184:                 --regexp='ubuntu.com'    |
185:         sort      --unique              >> "$OUTPUTFILE_2"; then
186:         true
187:     fi
188:
189:     number_of_lines=$(wc --lines < "$OUTPUTFILE_2")
190:     number_of_lines=$((number_of_lines - 2))
191:     footer="Dit bestand bevat $number_of_lines door de gebruiker toegevoegde \
192: pakketbronnen."
193:     printf '%b\n' "$DASHES\n$footer" >> "$OUTPUTFILE_2"
194: }
195:
196: create_file_installed_by_user() {
197:     local number_of_lines=0
198:     local header="Deze pakketten zijn meestal geïnstalleerd met Software of \
199: met de opdracht 'sudo apt install <pakketnaam>'."
200:     local footer=''
201:
202:     printf '%s\n' "$text3 ..."
203:
204:     touch "$OUTPUTFILE_3" |& $LOGCMD
```

```
205:
206:     printf '%b\n' "$header\n$DASHES" > "$OUTPUTFILE_3"
207:
208:     # Tabell1 = Alle geïnstalleerde pakketten op dit moment
209:     #             (dpkg-query --show --showformat='${Package}\n' | sort)
210:     # Tabel2 = Alle *automatisch* geïnstalleerde pakketten op dit moment
211:     #             (apt-mark showauto | sort)
212:     # Tabel3 = Door gebruiker geïnstalleerd (Tabell1-Tabel2)
213:     #             (comm -13 Tabell1 Tabel2)
214:     (
215:         comm -23 \
216:         <(dpkg-query --show --showformat='${Package}\n' | sort) \
217:         <(apt-mark showauto | sort) \
218:     ) >> "$OUTPUTFILE_3"
219:     number_of_lines=$(wc --lines < "$OUTPUTFILE_3")
220:     number_of_lines=$((number_of_lines - 2))
221:     footer="Dit bestand bevat $number_of_lines door de gebruiker \
222: geïnstalleerde pakketten.
223: Meer info via de opdracht: apt-cache show <pakketnaam> | grep \
224: --regexp='Description:'"
225:     printf '%b\n' "$DASHES\n$footer" >> "$OUTPUTFILE_3"
226: }
227:
228: create_file_favorites() {
229:     local number_of_lines=0
230:     local header="Deze favorieten zijn de snelkoppelingen te bereiken via een \
231: (soort van) (snel)menu."
232:     local footer=''
233:
234:     printf '%s\n' "$text4 ..."
235:
236:     touch "$OUTPUTFILE_4" |& $LOGCMD
237:
238:     printf '%b\n' "$header\n$DASHES" > "$OUTPUTFILE_4"
239:
240:     if gsettings get org.gnome.shell favorite-apps &> /dev/null; then
241:         favorites=$(gsettings get org.gnome.shell favorite-apps)
242:     else
243:         printf '%s\n' "
244: favorieten veiliggesteld." >> "$OUTPUTFILE_4"
245:         favorites=''
246:     fi
247:     favorites=${favorites//,/}
248:     favorites=${favorites//.desktop/}
249:     favorites=${favorites//\[}/}
250:     favorites=${favorites//\]/}
251:     for favorite in $favorites; do
252:         printf '%s\n' "${favorite//"/} " >> "$OUTPUTFILE_4"
253:     done
254:     number_of_lines=$(wc --lines < "$OUTPUTFILE_4")
255:     number_of_lines=$((number_of_lines - 2))
256:     footer="Dit bestand bevat $number_of_lines favorieten."
257:     printf '%b\n' "$DASHES\n$footer" >> "$OUTPUTFILE_4"
258: }
259:
260: create_file_snaps() {
261:     local number_of_lines=0
262:     local header="Deze snaps zijn geïnstalleerd met '[sudo] install SNAP' of \
263: met Software."
264:     local footer=''
265:
266:     printf '%s\n' "$text5 ..."
267:
268:     touch "$OUTPUTFILE_5" |& $LOGCMD
269:
270:     printf '%b\n' "$header\n$DASHES" > "$OUTPUTFILE_5"
271:
272:     if ! snap list 2> /dev/null >> "$OUTPUTFILE_5"; then
```

```
273:     printf '%s\n' 'snap is niet geïnstalleerd op dit systeem' >> \  
274: "$OUTPUTFILE_5"  
275:     fi  
276:     number_of_lines=$(wc --lines < "$OUTPUTFILE_5")  
277:     number_of_lines=$((number_of_lines - 3))  
278:     footer="Dit bestand bevat $number_of_lines snaps."  
279:     printf '%b\n' "$DASHES\n$footer" >> "$OUTPUTFILE_5"  
280: }  
281:  
282: term_script() {  
283:     info "De vijf aangemaakte bestanden ($PROGRAMME-$HOSTNAME-*) staan in de  
284: Persoonlijke map van de superuser (/root).  
285: Deze bestanden kunnen gebruikt worden om de volledigheid van de  
286: installatie, later door kzinstall, te controleren."  
287:     exit $SUCCESS  
288: }  
289:  
290: # -----  
291: # Main line  
292: # -----  
293: main() {  
294:     init_script  
295:     check_input "$@"  
296:     process_input  
297:     term_script  
298: }  
299:  
300: main "$@"  
301:  
302: # EOF
```