

# Outil de recherche dans l'historique d'apt



Mise à jour le : 08/06/2023



Non fonctionnel avant cette date !

- Outil de recherche dans l'historique d'apt.
  - Recherche par motif de nom de paquet
  - Recherche par la date
  - Recherche par la commande d'appel d'apt
  - Recherche par le type d'action d'apt
  - Recherche possible dans les fichiers de logs archivés
  - Affichage humainement lisible
- Script directement fonctionnel (pas de modification nécessaire).



Nécessite *gawk* et *zcat* :

```
apt install gawk gzip
```

## L'aide

Apt History Research - ahr - 2023-06-08-22:05

Usage :

```
ahr [-a | -z N] [--apt [-0]] [-c COMMAND] [-d DATE] [-t TYPE] MOTIF
ahr -c COMMAND [-a | -z N] [--apt [-0]] [-d DATE] [-t TYPE] [MOTIF]
ahr -d DATE [-a | -z N] [--apt [-0]] [-c COMMAND] [-t TYPE] [MOTIF]
ahr -t TYPE [-a | -z N] [--apt [-0]] [-c COMMAND] [-d DATE] [MOTIF]
```

Outil de recherche dans l'historique d'apt.

Recherche sur MOTIF ou COMMAND ou DATE ou TYPE.

Si plusieurs champs de recherche sont indiqués, alors applique un ET.

Options :

-a, --all Rechercher dans tout l'historique, même archivé.  
Incompatible avec '-z'.

-c, --command COMMAND Rechercher sur la commande COMMAND.

-d, --date DATE Rechercher sur la date DATE affinée avec  
l'heure.

Format : AAAA[-MM[-JJ[ HH[:MM[:SS]]]]]

-h, --help	Afficher cette aide et quitter.
-t, --type TYPE	Rechercher selon le TYPE, TYPE prend les valeurs :
	Install, Reinstall, Upgrade, Remove et Purge
-z, --dezip N	Rechercher dans le fichier archive N : /var/log/apt/history.log.N.gz Incompatible avec '-a'.
-v, --version	Afficher la version et quitter.
--apt	N'afficher que la liste des noms des paquets. Le retour est utilisable par apt ou dpkg. Un nom de paquet par ligne.
-0	Seulement avec '--apt'. Utiliser le caractère NULL pour séparateur de nom de paquet et pas le retour à la ligne.

MOTIF est une partie ou un nom exact de paquet.  
COMMAND est une partie ou la commande appelant l'action d'apt.  
DATE est une partie ou la date de début de l'action d'apt.  
TYPE est le type d'action d'apt.  
Tawal®©

## Le script

(mis à jour le 08/06/2023 22:05) non fonctionnel avant cette date !

[ahr](#)

```
#!/bin/bash

# Nom      : ahr      Apt History Research
#
# Par      : Tawal
#
# Nécessite : gawk (apt install gawk)
#
version=2023-06-08-22:05
### Répertoire des logs d'apt (personnalisable)
apt_folder="/var/log/apt"

### Formats Textes
Surb='\e[1m'      # Surbrillance
Ital='\e[3m'      # Italique
```

```

Rst='\e[m'          # Reset

### Expressions régulières
# Entier (option -z)
rgx_int="^[[:digit:]]+$"
# Format de DATE (option -d.)
rgx_date="^[[:digit:]]{4}($|- (0[1-9]|1[0-2])($|-
(0[1-9]|[1-2][0-9]|3[0-1])($| ([0-1][0-9]|2[0-3])($|:[0-5][0-9]))))$"

### Fonctions
usage_quit()
{
    echo -e "Apt History Research - ahr - $version
Usage :
    ${Surb}${0##*/}      [-a | -z N] [--apt [-0]] [-c COMMAND] [-d DATE]
    [-t TYPE] MOTIF${Rst}
    ${Surb}${0##*/}      -c COMMAND [-a | -z N] [--apt [-0]]      [-d DATE]
    [-t TYPE] [MOTIF]${Rst}
    ${Surb}${0##*/}      -d DATE      [-a | -z N] [--apt [-0]] [-c COMMAND]
    [-t TYPE] [MOTIF]${Rst}
    ${Surb}${0##*/}      -t TYPE      [-a | -z N] [--apt [-0]] [-c COMMAND]
    [-d DATE] [MOTIF]${Rst}

Outil de recherche dans l'historique d'apt.
Recherche sur ${Surb}MOTIF${Rst} ou ${Surb}COMMAND${Rst} ou
${Surb}DATE${Rst} ou ${Surb}TYPE${Rst}.
Si plusieurs champs de recherche sont indiqués, alors applique un ET.

Options :
    -a, --all                Rechercher dans tout l'historique, même
archivé.
                                Incompatible avec '${Ital}-z${Rst}'.

    -c, --command ${Surb}COMMAND${Rst} Rechercher sur la commande
${Surb}COMMAND${Rst}.

    -d, --date ${Surb}DATE${Rst}      Rechercher sur la date
${Surb}DATE${Rst} affinée avec l'heure.
                                Format : ${Surb}AAAA[-MM[-JJ[
HH[:MM[:SS]]]]]${Rst}

    -h, --help                Afficher cette aide et quitter.

    -t, --type ${Surb}TYPE${Rst}      Rechercher selon le
${Surb}TYPE${Rst}, ${Surb}TYPE${Rst} prend les valeurs :
                                ${Surb}Install${Rst},
                                ${Surb}Reinstall${Rst}, ${Surb}Upgrade${Rst}, ${Surb}Remove${Rst} et
                                ${Surb}Purge${Rst}

    -z, --dezip ${Surb}N${Rst}        Rechercher dans le fichier
archive ${Surb}N${Rst} :

```

```

    ${Ital}$apt_folder/history.log.${Surb}N${Rst}${Ital}.gz${Rst}
        Incompatible avec '${Ital}-a${Rst}'.

    -v, --version                Afficher la version et quitter.

    --apt                        N'afficher que la liste des noms des
paquets.
                                Le retour est utilisable par
    ${Ital}apt${Rst} ou ${Ital}dpkg${Rst}.
                                Un nom de paquet par ligne.

    -0                            Seulement avec '${Ital}--apt${Rst}'.
                                Utiliser le caractère NULL pour séparateur
                                de nom de paquet et pas le retour à la
ligne.

    ${Surb}MOTIF${Rst} est une partie ou un nom exact de paquet.
    ${Surb}COMMAND${Rst} est une partie ou la commande appelant l'action
d'apt.
    ${Surb}DATE${Rst} est une partie ou la date de début de l'action d'apt.
    ${Surb}TYPE${Rst} est le type d'action d'apt."

    printf "%80s\n" "Tawal©©"
    exit 0
}

version_quit()
{
    echo "$version"
    exit
}

erreur_quit()
{
    echo -n "${0##*/} : "
    [ "$1" = "log" ]      && echo -e "Le fichier
'$apt_folder/history.log' n'existe pas."
    [ "$1" = "zip" ]      && echo -e "Le fichier
'$apt_folder/history.log.$2.gz' n'existe pas."
    [ "$1" = "opt" ]      && echo -e "Option ${Surb}'$2'${Rst} non-
reconnue."
    [ "$1" = "opt0" ]     && echo -e "Option '${Surb}-0${Rst}' utilisée
sans '${Surb}--apt${Rst}'."
    [ "$1" = "motif" ]    && echo -e "Motif absent ou pas d'utilisation
d'une des options '${Surb}-c${Rst}' ou '${Surb}-d${Rst}' ou '${Surb}-
t${Rst}'."
    [ "$1" = "opt_az" ]   && echo -e "Options '${Surb}-a${Rst}' et
 '${Surb}-z${Rst}' incompatibles."
    [ "$1" = "opt_z" ]    && echo -e "L'argument de l'option ${Surb}'-
z'${Rst} doit être un nombre."
}
```

```

[ "$1" = "opt_t" ]      && echo -e "L'option ${Surb}-t${Rst} ne prend
que les mots ${Surb}Install${Rst}, ${Surb}Reinstall${Rst},
${Surb}Upgrade${Rst}, ${Surb}Remove${Rst} et ${Surb}Purge${Rst}."
[ "$1" = "opt_d" ]      && echo -e "Date incorrecte : ${*:2}"
[[ $1 =~ gawk|zcat ]] && echo -e "Nécessite
${Ital}${BASH_REMATCH[0]}${Rst} : apt install ${BASH_REMATCH[0]}"

exit 1
} >&2

test_opt()
{
[ "$1" = "opt_t" ] && [ "$2" != "Install" ]      && \
[ "$2" != "Remove" ] && [ "$2" != "Upgrade" ] && \
[ "$2" != "Reinstall" ] && [ "$2" != "Purge" ] && return 1
[ "$1" = "opt_d" ] && ! [[ "$2" =~ $rgx_date ]] && return 1
[ "$1" = "opt_z" ] && ! [[ "$2" =~ $rgx_int ]] && return 1
return 0
}

input2treat()
{
local i f n zap="$1"
declare -a LogFich idx
shift

if [ "$1" = "all" ]
then
for f in "$apt_folder"/hist*gz
do
n="${f%.*}"; n="${n##*.}"
LogFich[n]="$f"
done
# shellcheck disable=SC2206
idx=( ${!LogFich[@]} )
for ((i=${#idx[@]}-1; i>=0; i--))
do
zcat "${LogFich[${idx[i]}]}"
done
[ ! "$zap" ] && cat "$apt_folder"/history.log

elif [ "$1" = "one" ]
then
zcat "$apt_folder"/history.log."$2".gz
else
cat "$apt_folder"/history.log
fi
}

### Gestion des Options.
shopt -s extglob nullglob

```

```
while getopts :ac:d:ht:vz:-:0 option
do
    case $option in
        a) opt_a="on"
           [ "$opt_z" ] && erreur_quit opt_az
           dezip_type="all"
           ;;
        c) opt_c="$OPTARG"
           ;;
        d) opt_d="$OPTARG"
           test_opt opt_d "$opt_d" || erreur_quit opt_d "$opt_d"
           ;;
        z) opt_z="on"
           [ "$opt_a" ] && erreur opt_az
           test_opt opt_z "$OPTARG" || erreur_quit opt_z
           dezip_type="one"
           dezip_nb="$OPTARG"
           ;;
        h) usage_quit
           ;;
        t) opt_t="$OPTARG"
           test_opt opt_t "$opt_t" || erreur_quit opt_t
           ;;
        v) version_quit
           ;;
        0) opt_null="on"
           ;;
        -) case $OPTARG in
            all) opt_a="on"
                 [ "$opt_z" ] && erreur_quit opt_az
                 dezip_type="all"
                 ;;
            apt) opt_apt="on"
                 ;;
            command) opt_c="${!OPTIND}"
                     OPTIND=$((OPTIND+1))
                     ;;
            date) opt_d="${!OPTIND}"
                 test_opt opt_d "$opt_d" || erreur_quit opt_d
                 OPTIND=$((OPTIND+1))
                 ;;
            dezip)
                 opt_z="on"
                 [ "$opt_a" ] && erreur opt_az
                 test_opt opt_z "${!OPTIND}" || erreur_quit opt_z
                 dezip_type="one"
                 dezip_nb=${!OPTIND}
                 OPTIND=$((OPTIND+1))
                 ;;
            help) usage_quit
        esac
    esac
done
```

```
;;
type) opt_t=${!OPTIND}
test_opt opt_t "$opt_t" || erreur_quit opt_t
OPTIND=$((OPTIND+1))
;;
version) version_quit
;;
*) erreur_quit opt --"$OPTARG"
;;
esac
;;
*) erreur_quit opt -"$OPTARG"
;;
esac
done
shift $((OPTIND-1))

if ! hash gawk 2>/dev/null
then
    erreur_quit gawk
fi

if ! hash zcat 2>dev/null
then
    erreur_quit zcat
fi

if [ "$1" ]
then
    rech="$1"
elif [ ! "$opt_c" ] && [ ! "$opt_d" ] && [ ! "$opt_t" ]
then
    erreur_quit motif
fi

if [ "$opt_null" ] && [ ! "$opt_apt" ]
then
    erreur_quit opt0
fi

if [ "$opt_z" ]
then
    [ -e "$apt_folder"/history.log."$dezip_nb".gz ] || erreur zip
"$dezip_nb"
else
    if ! [ -e "$apt_folder"/history.log ]
    then
        if [ "$opt_a" ]
        then
            zap="on"
        else
```

```
erreur_quit log

fi
fi
fi

### Corps du programme
# shellcheck disable=SC1004
gawk -vrech="$rech" -vopt_c="$opt_c" -vopt_d="$opt_d" -vopt_t="$opt_t"
-vapt="$opt_apt" -vnull="$opt_null" '
function afficher(verb, zero, mess, Array, ArrayTmp, tmp, k,n) {
    if (0 in Array) {
        if (!verb && mess) {print mess}
        for (n in Array) {
            if (tmp && Array[n] ~ /,/ ) {
                ArrayTmp[k++]=Array[n]
                continue
            }
            if (!verb) {printf("\t")}
            printf(Array[n])
            if (zero) {printf("\0")}else{printf("\n")}
        }
    }
}
BEGIN {
    PROCINFO["sorted_in"]="@val_str_asc"
}
{
    if ($1 ~ /Start/) {
        start="Date Début : "$2" "$3
        date=$2" "$3
        ok=0
        next
    }
    if ($1 ~ /Commandline/) {
        command=gensub($1" ", "", 1)
        next
    }
    if ($1 ~ /Requested-By/) {
        demand=gensub($1" ", "", 1)
        next
    }
    if ($1~ /End/) {
        end="Date Fin : "$2" "$3
        ok=1
    }
    if (opt_c && command ~ opt_c) {opt_c_ok=1}
    if (opt_d && date ~ opt_d) {opt_d_ok=1}
    if (opt_t && $1 ~ opt_t) {opt_t_ok=1}
    if (rech && $0 ~ rech) {rech_ok=1}
```



```

if ((opt_c_ok && opt_d_ok && rech_ok && opt_t_ok) || \
    (!opt_c && opt_d_ok && rech_ok && opt_t_ok) || \
    (!opt_c && !opt_d && rech_ok && opt_t_ok) || \
    (opt_c_ok && !opt_d && rech_ok && opt_t_ok) || \
    (opt_c_ok&& opt_d_ok && !rech && opt_t_ok) || \
    (!opt_c && opt_d_ok && !rech && opt_t_ok) || \
    (opt_c_ok && !opt_d && !rech && opt_t_ok) || \
    (opt_c_ok && opt_d_ok && rech_ok && !opt_t) || \
    (!opt_c && opt_d_ok && rech_ok && !opt_t) || \
    (!opt_c && !opt_d && rech_ok && !opt_t) || \
    (opt_c_ok && !opt_d && rech_ok && !opt_t) || \
    (opt_c_ok && opt_d_ok && !rech && !opt_t) || \
    (!opt_c && opt_d_ok && !rech && !opt_t) || \
    (opt_c_ok && !opt_d && !rech && !opt_t) || \
    (!opt_c && !opt_d && !rech && opt_t_ok)) {
    ok1=1
}
opt_c_ok=0; opt_d_ok=0; opt_t_ok=0; rech_ok=0
if (ok1) {
    for (v=2; v<=NF-1; v++) {
        if ($v !~ /,/){
            if (apt) {
                paq=$v
            }else{
                if ($(v+2) ~ /)/) {
                    der=gensub(","," ", 1, $(v+2))
                    paq=$v" "$$(v+1)" "der
                }else{
                    pre=gensub(","," ", 1, $(v+1))
                    paq=$v" "pre
                }
            }
            if ($1 ~ /Install:/) {
                I[a++]=paq
            }else if ($1 ~ /Upgrade:/) {
                U[b++]=paq
            }else if ($1 ~ /Remove:/) {
                R[c++]=paq
            }else if ($1 ~ /Purge:/) {
                P[d++]=paq
            }else if ($1 ~ /Reinstall:/) {
                Re[e++]=paq
            }
        }
    }
}
if (ok && ok1) {
    if (!apt) {
        printf("\n")
        print start
        print end
    }
}

```

```
        if (command) {print "Commande      : "command}
        if (demand) {print "Demandeur    : "demand}
    }

    afficher(apt, null, "Installés :", I, Itmp, "on")
    afficher(apt, null, "", Itmp)
    afficher(apt, null, "Upgradés :", U)
    afficher(apt, null, "Désinstallés :", R)
    afficher(apt, null, "Purgés :", P)
    afficher(apt, null, "Réinstallés :", Re)

    if (!apt) {
        if (a>1) {s1="s"}else{s1=""}
        if (b>1) {s2="s"}else{s2=""}
        if (c>1) {s3="s"}else{s3=""}
        if (d>1) {s4="s"}else{s4=""}
        if (e>1) {s5="s"}else{s5=""}
        if (a>0) {print a,"paquet"s1" installé"s1}
        if (b>0) {print b,"paquet"s2" upgradé"s2}
        if (c>0) {print c,"paquet"s3" désinstallé"s3}
        if (d>0) {print d,"paquet"s4" purgé"s4}
        if (e>0) {print e,"paquet"s5" réinstallé"s5}
        printf("\n")
    }
    delete I; delete U; delete R; delete P; delete Re; delete

Itmp

    command=""
    date=""
    demand=""
    a=0; b=0; c=0; d=0; e=0; ok=0; ok1=0
}

' < <(input2treat "$zap" "$dezip_type" "$dezip_nb")
```

## Exemples d'utilisation et de retours

### Afficher tout l'historique (archives comprises)

```
ahr -a .
```

### Afficher l'historique non archivé concernant un paquet

```
ahr youtube-dl
```

```
Date Début   : 2023-03-02 09:55:13
```

```
Date Fin      : 2023-03-02 09:55:57
Commande      : apt -t bullseye-backports install youtube-dl
Demandeur     : tawal (1000)
Upgradés :
  youtube-dl:amd64 (2021.06.06-1, 2021.12.17-1~bpo11+1)
1 paquet upgradé
```

## Afficher l'historique du mois de mars 2023

```
ahr -ad 2023-03
```

```
Date Début   : 2023-03-01 22:28:53
Date Fin     : 2023-03-01 22:29:14
Commande     : apt purge linux-image-5.10.0-19-amd64
Demandeur    : tawal (1000)
Purgés :
  linux-image-5.10.0-19-amd64:amd64 ( )
1 paquet purgé
```

```
Date Début   : 2023-03-02 09:47:51
Date Fin     : 2023-03-02 09:49:38
Commande     : apt install yt-dlp
Demandeur    : tawal (1000)
Installés :
  yt-dlp:amd64 (2023.02.17-1~bpo11+1)
  python3-brotli:amd64 (1.0.9-2+b2, automatic)
  python3-mutagen:amd64 (1.45.1-2, automatic)
  python3-pycryptodome:amd64 (3.9.7+dfsg1-1+b2, automatic)
  python3-websockets:amd64 (8.1-1, automatic)
5 paquets installés
```

```
Date Début   : 2023-03-02 09:55:13
Date Fin     : 2023-03-02 09:55:57
Commande     : apt -t bullseye-backports install youtube-dl
Demandeur    : tawal (1000)
Upgradés :
  youtube-dl:amd64 (2021.06.06-1, 2021.12.17-1~bpo11+1)
1 paquet upgradé
```

## Afficher l'historique des désinstallations d'unattended-upgrade dans le fichier archive 1

```
ahr -c unattended -t Remove -z1
```

```
Date Début   : 2023-04-29 23:34:02
Date Fin     : 2023-04-29 23:34:23
Commande     : /usr/bin/unattended-upgrade
```

Demandeur : tawal (1000)  
Désinstallés :  
    linux-headers-5.10.0-20-amd64:amd64 (5.10.158-2)  
1 paquet désinstallé

Date Début : 2023-04-29 23:34:38  
Date Fin : 2023-04-29 23:34:48  
Commande : /usr/bin/unattended-upgrade  
Désinstallés :  
    linux-headers-5.10.0-20-common:amd64 (5.10.158-2)  
1 paquet désinstallé

Date Début : 2023-04-29 23:35:05  
Date Fin : 2023-04-29 23:35:41  
Commande : /usr/bin/unattended-upgrade  
Désinstallés :  
    linux-image-5.10.0-20-amd64:amd64 (5.10.158-2)  
1 paquet désinstallé

From:  
<http://debian-facile.org/> - **Documentation - Wiki**

Permanent link:  
<http://debian-facile.org/utilisateurs:tawal:scripts:apt-history-research>

Last update: **08/06/2023 22:38**

