

dwhy manual

ROBERT LUCKY wrote¹ about his computer: “I have no idea what programs and data are in there anymore.” And I was sure he was unlucky for not using Debian, because with Debian you can learn why each and every program and file is in the computer. To prove it, I wrote `dwhy`.

1 Syntax

Dash script `dwhy` uses syntax: `dwhy name`. It tells me why the object named `name` is in my Debian system. Object `name` can be a file, or a program, or a package.

2 Examples

I will explain `dwhy` showing some examples.

2.1 File

I will learn first why file `/etc/crontab` is in my system.

```
$ dwhy /etc/crontab
/etc/crontab: ASCII English text
Package: pool/main/c/cron/cron_3.0p11-116_amd64.deb [important]
cron (M) <- base system (D)
```

So `dwhy` tells me that `/etc/crontab` is a text file, which is part of package `cron`, version `3.0p11-116`, for the architecture `amd64`, and priority `important`. Finally, it tells me that package `cron` was manually (M) installed as part of Debian (D) base system.

2.2 Program

Now, I will try `dwhy` with a program, for example `file`.

```
$ dwhy file
file -> /usr/bin/file
/usr/bin/file: ELF 64-bit LSB executable, x86-64, [...]
Package: pool/main/f/file/file_5.04-5_amd64.deb [standard]
file (M) <- standard system (D)
```

Now, `dwhy` first locates the executable file, `/usr/bin/file`, and then explains that it is part of package `file`, priority `standard`, installed by Debian (D) as part of the standard system.

¹ “Computer Rot” by Robert Lucky, on IEEE Spectrum, July 2007, available in his page <http://www.boblucky.com/reflect/july07.htm>.

2.3 Package

If I apply `dwhy` to a package installed in my system, for example `samba`, the session goes as follows:

```
$ dwhy samba
samba: file not found, try as package
Package: pool/main/s/samba/samba_3.5.6~[...]._amd64.deb [optional]
samba (M)
```

It means that I have installed package `samba` myself manually (M).

2.4 Directory

If I use a directory as argument, I also get some information.

```
$ dwhy /etc/network
/etc/network: directory
Package: pool/main/n/netbase/netbase_4.45_all.deb [important]
netbase (M) <- base system (D)
Package: pool/main/w/wpasupplicant/wpasupplicant_[...].deb [optional]
wpasupplicant (A) <- network-manager-gnome network-manager
network-manager-gnome (A)
network-manager (A) <- network-manager-gnome (A)
Package: pool/main/i/ufw/ufw_0.6.10_amd64.deb [important]
ufw (M) <- base system (D)
Package: pool/main/v/vde2/vde2_2.2.3-3_amd64.deb [optional]
vde2 (A)
Package: pool/main/s/samba/samba_3.5.6~[...]._amd64.deb [optional]
samba (M)
Package: pool/main/o/openssh/openssh-server_[...]._amd64.deb [optional]
openssh-server (M) <- ssh-server (T)
Package: pool/main/s/sysvinit/initscripts_[...]._amd64.deb [required]
initscripts (M) <- base system (D)
Package: pool/main/a/avahi/avahi-daemon_[...]._amd64.deb [optional]
avahi-daemon (M) <- desktop (T)
```

The directory is used by several packages, so `dwhy` has to explain why each one is installed. Here I get all kind of explanations. Package `wpasupplicant` was installed automatically (A), either by package `network-manager-gnome` or package `network-manager`, so `dwhy` investigates each one. Package `vde` was installed automatically (A), though it does not depend on any other package; this means that it was installed as a recommendation. Package `openssh-server` was installed as part of the task (T) `ssh-server`.

2.5 Out of Debian

Of course, Debian is not responsible of every file in my system.

```
$ dwhy dwhy
/home/user/src/dash/dwhy/dwhy: a /bin/dash script text executable
dpkg: /home/user/src/dash/dwhy/dwhy not found.
/home/user/src/dash/dwhy/dwhy is not in a Debian package
```

And, of course, I have not installed every Debian package.

```
$ dwhy kde
kde: file not found, try as package
No packages found matching kde.
Package: pool/main/m/meta-kde/kde_66_all.deb []
No packages found matching kde.
kde (X)
```

It means that package `kde` is not (X) installed in my system.

Finally, of course, sometimes I fail to ask a proper question.

```
$ dwhy foo
foo: file not found, and it is not a package.
```

3 Problems

One of the problems was already presented: `dwhy` cannot explain why packages installed as recommendations are in the system.

Another problem is that some files installed by Debian, as `/etc/fstab`, are not part of any package, though they are built by a package installation script, and `dwhy` does not do a good job with these.

```
$ dwhy /etc/fstab
/etc/fstab: ASCII English text
dpkg: /etc/fstab not found.
/etc/fstab is not in a Debian package
```

Speed can become a problem. I wrote `dwhy` on the basis of other programs, `dpkg-query`, `apt-cache`, `aptitude`, etc., instead of going straight to the databases in `/var/lib/dpkg`, `/var/lib/apt`, and `/var/lib/aptitude`. This can be ammended.

4 Links

You can get the last version of `dwhy` from <https://github.com/ramoncasares/dwhy>.

5 The code

Magicians advise not to reveal the tricks, but Debian is not magic.

dwhy

```
1 #!/bin/dash
2
3 # debwhy by www.ramoncasares.com 2011
4 # License: GPL
5 # Version: 0.1 (20110323)
6
7 dwhy () {
8   if test "$1" = "" ; then
9     echo "Enter: dwhy filename"
10    echo "          to let Debian explain why filename is in your system"
11    exit 1
12  fi
13
14  if test -e "$1" ; then
15    TARGET="$(readlink -f "$1")"
16    if test -L "$1" ; then
17      echo "$1 -> $TARGET"
18    fi
19  else
20    FIRSTTARGET="$(which "$1")"
21    TARGET="$(readlink -f "$FIRSTTARGET")"
22    if test -L "$FIRSTTARGET" ; then
23      echo "$1 -> $FIRSTTARGET -> $TARGET"
24    else
25      if test "$TARGET" != "" ; then
26        echo "$1 -> $TARGET"
27      fi
28    fi
29  fi
30
31  if test "$TARGET" = "" ; then
32    if test "[$(apt-cache -n search "^$1$")]" = "" ; then
33      echo "$1: file not found, and it is not a package."
34      exit 2
35    else
36      echo "$1: file not found, try as package"
37      PACKAGES=$1
38    fi
39  else
40    file "$TARGET"
41  fi
42
43  if test "$PACKAGES" = "" ; then
44    PACKAGES=$(dpkg-query --search "$TARGET" | cut -d: -f1 | sed s/,//g)
45  fi
46
47  if test "$PACKAGES" = "" ; then
48    echo "$TARGET is not in a Debian package"
49  else
50    for PACKAGE in $PACKAGES ; do
51      DFILE=$(apt-cache show "$PACKAGE" | grep "^Filename: " | cut -d' ' -f2)
```

```
52     PRIORITY=$(dpkg-query -W -f='${Priority}' $PACKAGE)
53     echo "Package: $DFILE [$PRIORITY]"
54     dtree $PACKAGE
55 done
56 fi
57
58 exit 0
59 }
60
61 dtasks () {
62     if test "[${ITASKS}] = []" ; then
63         ITASKS=$(tasksel --list-tasks | grep "^i" | sed 's/^i \([^\\t]*\)*/\1/')
64     fi
65     TASKS=""
66     for TASK in $ITASKS
67     do
68         if test "$(aptitude search "$1~t$TASK")" != "" ; then
69             TASKS="$TASKS $TASK"
70         fi
71     done
72 }
73
74 dtree () {
75 if test $# = 0 ; then
76     echo "Enter: dtree package1 package2 ..."
77     echo "        to draw the tree of packages and tasks of package1 ..."
78     return 1
79 fi
80
81 for PACKAGE in $@ ; do
82     PRIORITY=$(dpkg-query -W -f='${Priority}' $PACKAGE)
83     SEARCH="$(aptitude search "^$PACKAGE$")"
84     if test "[$(echo "$SEARCH" | cut -c1)]" = "[i]"
85     then
86         if test "[$(echo "$SEARCH" | cut -c3)]" = "[A]" ; then
87             STATUS="A"
88         else
89             STATUS="M"
90         fi
91     else
92         STATUS="X"
93     fi
94     echo -n " $PACKAGE ($STATUS)"
95     if test "$PRIORITY" = required -o "$PRIORITY" = important ; then
96         echo " <- base system (D)"
97     else
98         if test "$PRIORITY" = standard ; then
99             echo " <- standard system (D)"
100        else
101            if test "$STATUS" = "M" ; then
102                dtasks "$PACKAGE"
103                if test "[${TASKS}] != []" ; then
104                    echo -n " <-"
105                    echo "$TASKS (T)"
106                else
107                    echo ""
```

```
108         fi
109     else
110         RPKGS=$(apt-get -qq -s remove "$PACKAGE" \
111             | grep "^Remv" | cut -d' ' -f2 \
112             | grep -vx "$PACKAGE")
113         if test $(echo "$RPKGS" | wc -w) -eq 0 ; then
114             echo ""
115         elif test $(echo "$RPKGS" | wc -w) -eq 1 ; then
116             echo -n " <-"
117             dtree $RPKGS
118         else
119             echo -n " <- "
120             echo "$RPKGS" | tr "\n" " "
121             echo ""
122             dtree $(echo $RPKGS | tr "\n" " ")
123         fi
124     fi
125 fi
126 fi
127 done
128
129 return 0
130 }
131
132 dwhy $@
133
```