Rethinking of the debian/watch

With thought experiments about uscan

Kentaro Hayashi

DebConf18 in Taiwan

2018-08-03

ClearCode Inc.

Digest of this talk

- Current d/watch file is sometimes complicated
- Update to new format (v5) can solve it

Agenda

- Who I am?
- Why I started to play with debian/watch?
- Introduction about debian/watch
- The debian/watch current statistics
- Thought experiments about debian/watch
- Conclusion

Agenda

- **Who I am?**
- Why I started to play with debian/watch?
- Introduction about debian/watch
- The debian/watch current statistics
- Thought experiments about debian/watch
- Conclusion

Who I am?

- Kentaro Hayashi <kenhys@gmail.com>
- Twitter/GitHub (@kenhys) / Debian contributor (@kenhys-guest)
- **Trackpoint fan** - soft dome user
- Working for ClearCode Inc.

Ad: ClearCode Inc.

ClearCode

- <URL:https://www.clear-code.com/>
  - Free software is important in ClearCode Inc.
  - We develop/support software with our free software development experiences.
  - We feed back our business experiences to free software.

As a contributor

- Maintainer of some packages
  - groonga (Upstream releases monthly updates)
  - fcitx-imlist
  - libhinawa
  - <URL:https://qa.debian.org/developer.php?email=hayashi@clear-code.com>

Agenda

- Who I am?
- **Why I started to play with debian/watch?**
- Introduction about debian/watch
- The debian/watch current statistics
- Thought experiments about debian/watch
- Conclusion

Why playing with d/ watch?

- #899119: Need redirector for osdn.net
  - <URL:https://bugs.debian.org/cgi-bin/bugreport.cgi?bug=899119>

d/watch for fonts-sawarabi-mincho

```
version=4
opts="uversionmangle=s/-beta/~beta/;s/-rc/~rc/;s/-preview/~preview/, \
pagemangle=s%<osdn:file url="([^<]*)</osdn:file>%<a href="$1">$1</a>%g, \
downloadurlmangle=s%projects/sawarabi-fonts/downloads%frs/redir\.php?m=iij&f=sawarabi-fonts%g;s/xz\//xz/" \
https://osdn.net/projects/sawarabi-fonts/releases/rss \
https://osdn.net/projects/sawarabi-fonts/downloads/.*/sawarabi-mincho@ANY_VERSION@@ARCHIVE_EXT@/ debian uupdate
```

- Need to parse RSS!

d/watch for fonts-sawarabi-mincho

- Combination with:
  - pagemangle
  - downloadurlmangle
  - uversionmangle

pagemangle?

- pagemangle=s%<osdn:file url="([^<]*)</osdn:file>%<a href="$1">$1</a>%g,
  - Convert a page content
    - <osdn:file url="([^<]*)</osdn:file> ➡ <a href="$1">$1</a>

downloadurlmangle?

- downloadurlmangle=s%projects/sawarabi-fonts/downloads%frs/redir\.php?m=iij&f=sawarabi-fonts%g;s/xz\//xz/"
  - Convert a download url
    - projects/sawarabi-fonts/downloads ➡ frs/redir\.php?m=iij&f=sawarabi-fonts
    - xz/ ➡ xz

uversionmangle?

- uversionmangle=s/-beta/~beta/;s/-rc/~rc/;s/-preview/~preview/
  - Convert a specific suffix
    - -beta ➡ ~beta
    - -rc ➡ ~rc
    - -preview ➡ ~preview

#899119

> *Hideki Yamane:*
> *"They sometimes changes download way to reduce download accessby preventing bot, so debian/watch file is complicated and it annoyed us. Implementing redirector in qa.debian.org would improvethis situation."*

[「#899119#5」より引用]

Motivation

- It seems that sometimes d/watch file is **too complicated**
  - I'll look into d/watch a bit

Agenda

- Who I am?
- Why I started to play with debian/watch?
- **Introduction about debian/watch**
- The debian/watch current statistics
- Thought experiments about debian/watch
- Conclusion

Introduction about debian/watch

- Used to check for newer versions of upstream software
- https://wiki.debian.org/debian/watch is the good start point

The typical examples

- There are 8 examples
  - Bitbucket, GitHub, Gitlab(Salsa), Google Code, LaunchPad, PyPI, and Sourceforge

Common mistakes to avoid

- There are 8 common mistakes in d/watch
  - see: https://wiki.debian.org/debian/watch

Common mistakes(1)

- Not escaping dots, which match any character
- The solution is:
  - Use \. instead of . in the regex

Common mistakes(2)

- A file extension regex that is not flexible enough
- The solution is:
  - Use **\.(?:zip|tgz|tbz|txz|(?:tar\.(?:gz|bz2|xz)))**

Common mistakes(3)

- Not anchoring the version group at the right place
- The solution is:
  - Include something before (\d\S+) like fooproj-**(\d\S+)**\.tar\.gz

Common mistakes(4)

- Not starting the version part of the regex with a digit
- The solution is:
  - Use **\d** instead of **.**

Common mistakes(5)

- Not being flexible enough in the path to the file
- The solution is:
  - Use http://example.com/someproject/.*/program-(\d\S+)\.tar\.gz instead of http://example.com/someproject/**path/to/program/downloads**/program-(\d\S+)\.tar\.gz

Common mistakes(6)

- Not mangling upstream versions that are alphas, betas or release candidates to make them sort before the final release

- The solution is:
  - Use **uversionmangle** like opts=uversionmangle=s/(\d)[_\.\-\+]?((RC|rc|pre|dev|beta|alpha)\d*)$/$1~$2/

Common mistakes(7)

- Not mangling Debian versions to remove the +dfsg.1 or +dfsg1 suffix
- The solution is:
  - Use **dversionmangle** like opts=dversionmangle=s/\+(debian|dfsg|ds|deb)(\.?\d+)?$//

Common mistakes(8)

- Not enabling cryptographic signature verification when your upstream signs their releases with OpenPGP
- The solution is:
  - Support cryptographic signature!

Impression about d/watch

- It is okay once d/watch is prepared
- But, there are some pitfalls in d/watch

Motivation again

- d/watch is useful
- But too complicated
- It should be more simple! (somehow)

Agenda

- Who I am?
- Why I started to play with debian/watch?
- Introduction about debian/watch
- **The debian/watch current statistics**
- Thought experiments about debian/watch
- Conclusion

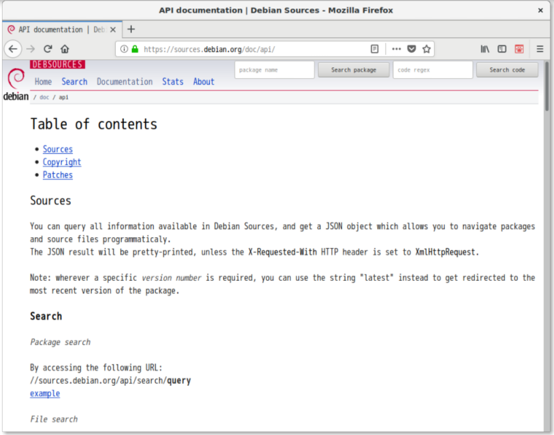
Why do we use statistics?

- We can't judge whether the idea is good or not
- Let's discuss based on **the fact (data)**

Collect d/watch data

- We have no data to judge
- But, we can use the API!
  - <URL:https://sources.debian.org/doc/api/>

sources.d.o API documentation



API documentation | Debian Sources - Mozilla Firefox
API documentation | Deb
https://sources.debian.org/doc/api/
DEBSOURCES
Home Search Documentation Stats About
package name
Search package
code regex
Search code
debian
/ doc / api

Table of contents

- Sources
- Copyright
- Patches

Sources

You can query all information available in Debian Sources, and get a JSON object which allows you to navigate packages and source files programmaticaly.
The JSON result will be pretty-printed, unless the X-Requested-With HTTP header is set to XmlHttpRequest.

Note: wherever a specific *version number* is required, you can use the string "latest" instead to get redirected to the most recent version of the package.

Search

Package search

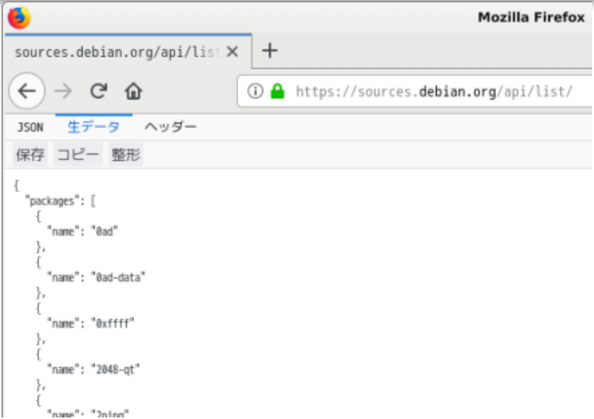
By accessing the following URL:
//sources.debian.org/api/search/**query**
example

File search

Collect package list

- Access package list API
  - <URL:https://sources.debian.org/api/list>
  - You can use this API to collect **source** package list

e.g. source package list



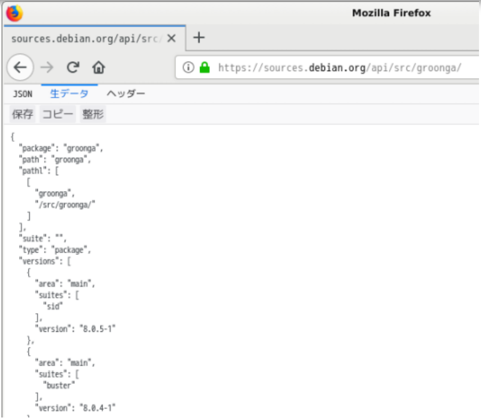
Mozilla Firefox
sources.debian.org/api/list
https://sources.debian.org/api/list/
JSON
生データ
ヘッダー
保存
コピー
整形

```
{
  "packages": [
    {
      "name": "0ad"
    },
    {
      "name": "0ad-data"
    },
    {
      "name": "0xffff"
    },
    {
      "name": "2048-qt"
    },
    {
      "name": "2ping"
```

Collect package info

- Access package info API
  - Get suites information about package
    - e.g. <URL:https://sources.debian.org/api/src/groonga/>
  - You can use this API to collect a specfic release package (e.g. collects sid only)

e.g. Groonga package info


Mozilla Firefox
sources.debian.org/api/src/
https://sources.debian.org/api/src/groonga/
JSON
生データ
ヘッダー
保存
コピー
整形

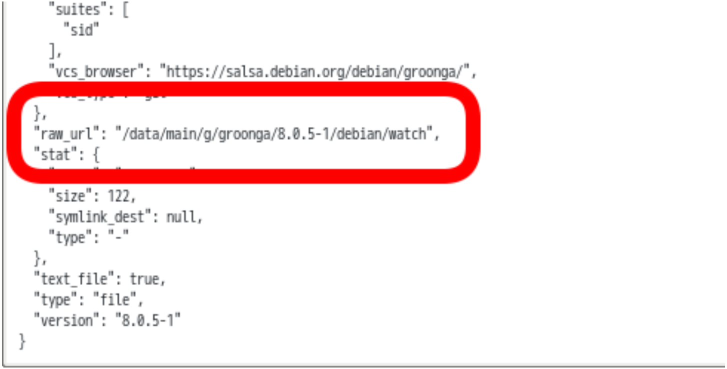
```
{
  "package": "groonga",
  "path": "groonga",
  "pathl": [
    [
      "groonga",
      "/src/groonga/"
    ]
  ],
  "suite": "",
  "type": "package",
  "versions": [
    {
      "area": "main",
      "suites": [
        "sid"
      ],
      "version": "8.0.5-1"
    },
    {
      "area": "main",
      "suites": [
        "buster"
      ],
      "version": "8.0.4-1"
```

Collect raw url

- Access file info API
  - Get path to raw url
    - e.g. <URL:https://sources.debian.org/api/src/groonga/latest/debian/watch/>

➡ https://sources.debian.org/api/src/groonga/**8.0.5-1**/debian/watch/

e.g. Groonga d/ watch raw url

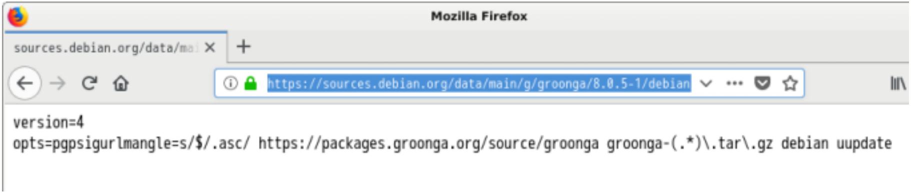


```
    "suites": [
      "sid"
    ],
    "vcs_browser": "https://salsa.debian.org/debian/groonga/",
  },
  "raw_url": "/data/main/g/groonga/8.0.5-1/debian/watch",
  "stat": {
    "size": 122,
    "symlink_dest": null,
    "type": "-"
  },
  "text_file": true,
  "type": "file",
  "version": "8.0.5-1"
}
```

Collect d/watch

- Access file content
  - Get raw content of d/watch
    - e.g. <URL:https://sources.debian.org/data/main/g/groonga/8.0.5-1/debian/watch>

e.g. Groonga d/watch



https://sources.debian.org/data/main/g/groonga/8.0.5-1/debian

```
version=4
opts=pgpsigurlmangle=s/$/.asc/ https://packages.groonga.org/source/groonga groonga-(.*)\.tar\.gz debian uupdate
```

We are ready to collect data

- Collect source package list in unstable (API)
- Collect each d/watch if available (API)
- Analyze and Visualize data (Task)

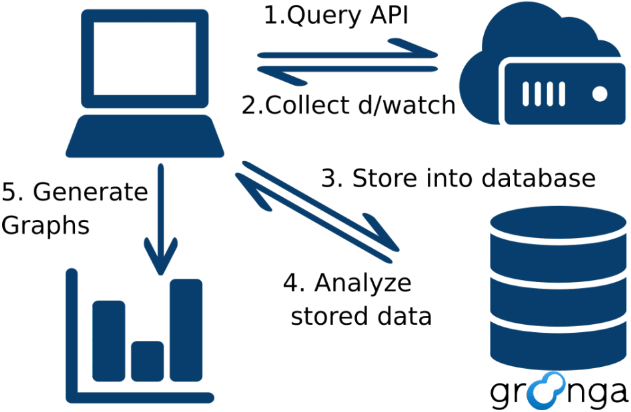
How to collect it?

- Use debsources-watch-crawler
  - <URL:https://github.com/kenhys/debsources-watch-crawler.git>
    - Crawling d/watch and store into database (using Groonga)

Parsing opts in d/watch

- Use Parse::Debian::Watch
  - <URL:https://github.com/kenhys/perl-Parse-Debian-Watch.git>
    - Extracted parser code from scripts/uscan.pl

Analyzing system components



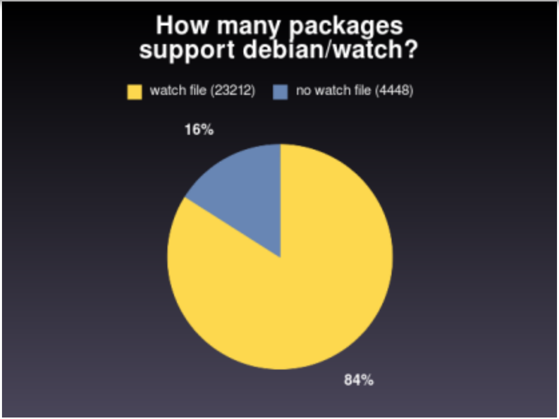
NOTE

- The data for statistics is snapshot at 2018/7
  - 39,074 source packages exists in debian
    - 27,660 unstable source packages

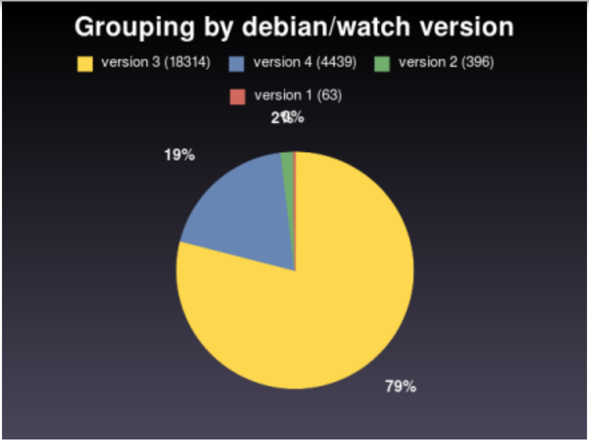
Some question about d/watch

- Is watch file used?
- Which version is used in package?
- What are the popular hosting sites?

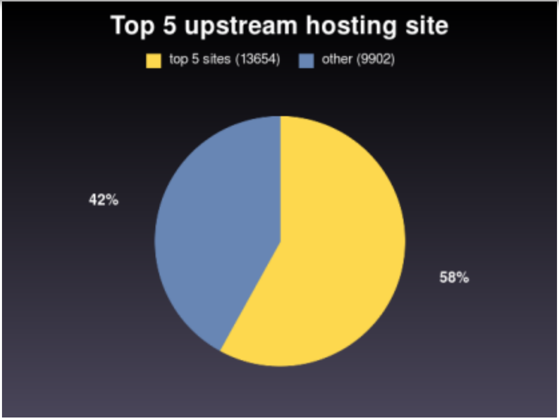
Is watch file used?



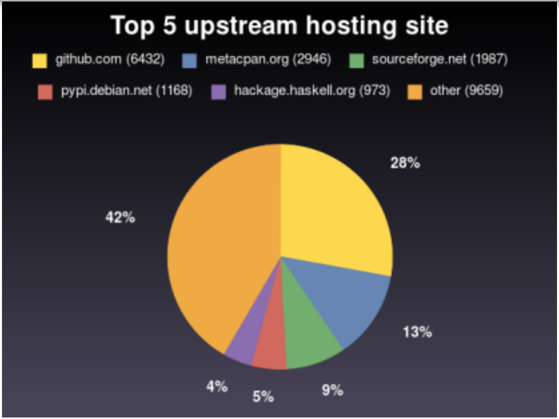
What version are you using?



Top 5 hosting covers 58%



Popular hosting?



These graphs show

- 84% source packages already support d/watch.
- It seems that there is a room for optimizing for top 5 hosting sites

What option is frequently used?

- Option is ...
  - Not used
  - Rarely used
  - Sometimes used
  - Often used

Not used option

- bare: 0
- nopasv: 0
- hrefdecode: 0
- pretty: 0
- unzipopt: 0

Rarely used

- user-agent: 3
- gitmode: 4
- dirversionmangle: 5
- date:9
- oversionmangle: 10

Rarely used (2)

- component: 13
- decompress: 18
- versionmangle: 11
- passive: 30
- pagemangle: 31

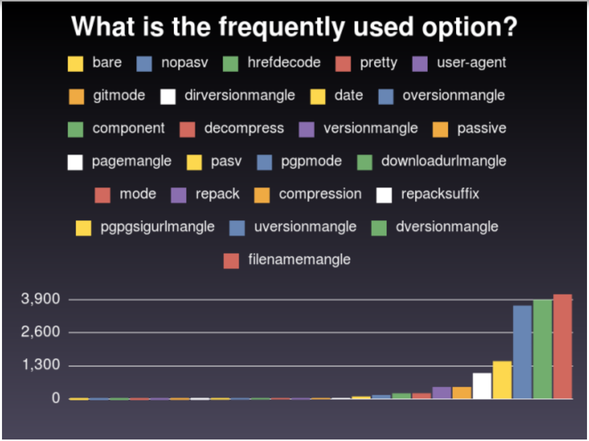
Sometimes used

- pasv: 120
- pgpmode: 175
- downloadurlmangle: 247
- mode: 249
- repack: 491
- compression: 489

Often used

- repacksuffix: 1039
- pgpsigurlmangle: 1510
- uversionmangle: 3695
- dversionmangle: 3921
- filenamemangle: 4134

What is the frequently used one?



Thought experiments d/watch

- The facts
  - Top 5 upstream hosting sites occupy 58%
  - Opts option usage is very limited
- The estimations
  - We can simplify d/watch by dropping support for not frequently used option

Required information?

- Some information to be parsed
  - Hosting
  - Owner
  - Project

The new syntax idea

- Some information to be parsed
  - Hosting ➡ type=...
  - Owner ➡ owner=...
  - Project ➡ project=...

e.g Diff between old and new rule

```
-version=4
+version=5

-opts=filenamemangle=s/.+\/v?(\d\S*)\.tar\.gz/fcitx-imlist-$1\.tar\.gz/
-  https://github.com/kenhys/fcitx-imlist/tags .*/v?(\d\S*)\.tar\.gz
+type=github.com,owner=kenhys,project=fcitx-imlist
```

e.g The new rule

```
version=5
type=github.com,owner=kenhys,project=fcitx-imlist
```

- e.g. <URL:https://github.com/kenhys/fcitx-imlist>

Pros

- for maintainer
  - Easy to maitain
  - It is flexible even though download url is changed (not domain change)
  - It avoids pitfalls by common mistakes which is listed in wiki.d.o

Cons

- for uscan developer
  - It needs to fix uscan for each hosting sites
    - The upstream uses minor hosting site, it can't migrate to the new rule until uscan supports
  - It may lack the functionality in contrast to existing rules
  - Traditinal and new style are needed to maitain

Experiments

- We don't know whether new rule is practical enough
- Let's do experiment!

Steps to verify

- 1. Modify uscan which supports new rule
- 2. Download the source package
- 3. Revert to the previous release for uscan
- 4. Uscan with current and modified rule
- 5. Compare **dehs** result

Dehs?

- Debian External Health Status
  - <URL:https://wiki.debian.org/DHES>
  - Machine readable output of uscan
    - It's easy to detect regression
    - Without regression, new rule has enough functionality!

Test case

- New rule for GitHub
  - The typical use case
- ~~New rule for OSDN~~
  - The minior use case
  - It needs more work (Currently in modified version, dehs output is broken)

The new rule for GitHub

```
version=5
type=github.com,owner=kenhys,project=fcitx-imlist
```

How to modify uscan

- Add a patch to scripts/uscan.pl
  - Bump version to 5
  - Add regular expression to parse a new rule
  - Assign mangle to $options to emulate
  - Repeat above steps to support more patterns
  - <URL:https://salsa.debian.org/kenhys-guest/devscripts/tree/add-type-rule>

How good enough new d/watch rule?

- DEMO
  - The new rule for fcitx-imlist (GitHub)

Conclusion

- There is a bit redundant case in d/watch
- d/watch can be simplified by new d/watch rule
  - But not fully verified yet. It needs more testing!
- Feedback is welcome!

Q. What about fakeupstream.cgi?

- fakeupstream.cgi returns only list of releases, so it is not useful to simplify the rule

Q. What about redirector?

- Yes, you are right. But it needs to be supported in server side and uscan side
- The new rule only requires to implemented in uscan