

Debianパッケージング repack編

そのソフトウェア、そのまま配布して大丈夫だっけ

Kentaro Hayashi

ClearCode Inc.

2022年11月 東京エリア・関西合同Debian勉強会





スライドはRabbit Slide Showにて公開済みです

- Debianパッケージング repack編
 - <https://slide.rabbit-shocker.org/authors/kenhys/tokyodebian-repack-howto-202211/>



本日の内容

- 自由なソフトウェア・不自由なソフトウェア
- バージョンによる区別のしかた(+dfsg or +ds)
- 再パッケージングどうやる問題



Debian(たまに)あるある

- 使いたいソフトウェアはまだaptでインストール不可
- そうだ、パッケージングしよう!!
 - 自由なソフトウェアである
 - 今回の話題の対象外
 - (Debian的には) **不自由なソフトウェアである** 😞



自由なソフトウェアでない

- いろんなパターンがある
 - 自由でないのが主要構成部分か
 - さすがにどうしようもないので諦める 🙄
 - 自由でないのは一部である
 - 例: rapidjson (JSONライセンス)
<https://wiki.debian.org/qa.debian.org/jsonevil>
 - バンドルしているソフトウェアが自由でない



自由でないのは一部である

- **問題のある部分を除去**してmainとして配布する



再配布できるようにしたソフトウェアの見分け方

- パッケージに+dfsg or +dsがついている
- 参考: lintianに新規ルールを追加する方法
<https://slide.rabbit-shocker.org/authors/kenhys/tokyodebian-lintian-howto-202112/>



+dfsg と +ds の区別

- +dfsg

- DFSGに適合しないものを除去した
- 例: rapidjson-dev 1.1.0+dfsg2-7.1

- +ds

- DFSGとの適合云々とは別の理由で除去した
- 例: バンドルしているソフトウェアを削除



再パッケージングのしかた

- 必要なもの
 - (パッケージ名)_(バージョン)+dfsg.orig.tar.gz
 - (パッケージ名)_(バージョン)+ds.orig.tar.gz
- +dfsgや+dsなアーカイブはどうつくる？



repackして除去

- debian/copyrightにFiles-Excluded:を書く
- debian/watchでrepackルールを記述する
 - debian/rulesでget-orig-sourceルールでrepackするのは古いやりかた
 - debian/以下のスクリプトを実行してアーカイブみたいなことをしなくてもいい



Files-Excluded:について

- Files-Excluded:の仕様に関するバグ
 - <https://bugs.debian.org/cgi-bin/bugreport.cgi?bug=685506>
- Files-Excluded:のドキュメント化に関するバグ
 - <https://bugs.debian.org/cgi-bin/bugreport.cgi?bug=1000771>
- <https://wiki.debian.org/UscanEnhancements>



debian/copyrightにどう書くか

Files-Excluded:

(削除対象のファイル・ディレクトリにマッチするパターンを列挙)

Comment:

(削除対象のファイル・ディレクトリに関する説明)



debian/copyrightの例

■ groonga-12.0.9+dfsg-1

```
Format: https://www.debian.org/doc/packaging-manuals/copyright-format/1.0/  
Upstream-Name: Groonga  
Upstream-Contact: Groonga Project <packages@groonga.org>  
Source: http://packages.groonga.org/source/groonga/  
Files-Excluded:  
doc/locale/en/html/*.html  
doc/locale/en/html/_static  
doc/locale/en/html/searchindex.js  
doc/locale/ja/html/*.html  
doc/locale/ja/html/searchindex.js  
doc/locale/ja/html/_static  
vendor/rapidjson-1.1.0  
Comment: The following files were removed because  
doc/locale/en/html/*.html: regenerated during building package  
...
```

Files: *



Files-Excluded:の注意

- **必ずヘッダパラグラフに書くこと**

- 以下のように改行入れると機能しなくてはまる

```
Format: https://www.debian.org/doc/packaging-manuals/copyright-format/1.0/  
Upstream-Name: Groonga  
Upstream-Contact: Groonga Project <packages@groonga.org>  
Source: http://packages.groonga.org/source/groonga/
```

```
Files-Excluded:  
doc/locale/en/html/*.html  
...  
vendor/rapidjson-1.1.0  
Comment: The following files were removed because  
doc/locale/en/html/*.html: regenerated during building package  
...
```

```
Files: *
```



debian/watchの例

- repackとrepacksuffix=+dfsgを指定する
 - やり直す場合repacksuffix=+dfsg2というように増えていく

```
version=4
opts=\
repack,\
dversionmangle=auto,\
repacksuffix=+dfsg,\
pgpsigurlmangle=s/$.asc/\
https://packages.groonga.org/source/@PACKAGE@/@PACKAGE@-@ANY_VERSION@@@ARCHIVE_EXT@
```



パッケージングの流れ (1)

- `gbp import-orig -uscan`
 - +dfsgなアーカイブが作成される
 - upstream ブランチにインポート
 - upstream/(バージョン)+dfsg タグ付与
 - pristine-tar ブランチに (パッケージ)_(バージョン)+dfsg.orig.tar.xz.(delta|id) が作成される



パッケージングの流れ (2)

- gbp dchでdebian/changelogを更新 & もろもろ作業
- debsign & dputでアップロード
 - dput-ngだとまとめてできる



repackしくじった場合

- upstream ブランチリセット
- pristine-tar リセット
- `git tag -d upstream/(バージョン)+dfsg`
- `gbp import-orig` やり直しなどちょっと面倒くさい
 - `gbp pq rebase` で盛大にコンフリクト...
 - うっかり `patch-queue` 飛ばして泣く (事故には気をつけよう)



さいごに

- 部分的に不自由なソフトウェアを再配布可能にするには
 - d/copyrightでFiles-Excluded:を使え
 - d/watchでrepack,repacksuffixを記述