

# .Debian

銀河系唯一のDebian専門誌

2015年1月17日

特集：Emacs関連パッケージの作成



# 第 121 回 Debian 勉強会

---

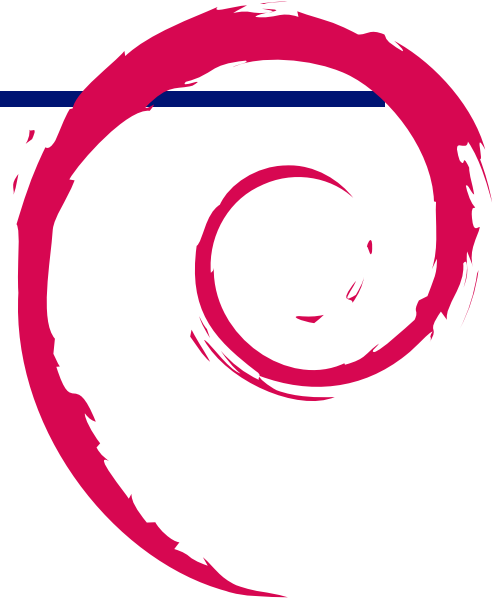
## 目次

1	事前課題	2	4	Emacs 関連パッケージの Debian パッケージ作成	6
1.1	野島	2	4.1	はじめに	6
1.2	alohaug	2	4.2	step by step での作り方	6
1.3	henrich	2	4.3	まとめ	9
1.4	koedoyoshida	2	5	会場での無線 LAN のつなぎ方	10
1.5	dictoss	2	5.1	はじめに	10
1.6	yyatsuo	2	5.2	wpasupplicant 及び /etc/network/interfaces を利用の場合	10
1.7	yy-y-ja.jp	3	5.3	その他の無線 LAN 用パッケージを利用の場合	10
1.8	Roger Yan	3			
2	Debian Trivia Quiz	4			
3	最近の Debian 関連のミーティング報告	5			
			3.1	第 121 回東京エリア Debian 勉強会	5

---

# 1 事前課題

野島 貴英



今回の事前課題は以下です:

1. 本日、何の作業をやるかを宣言ください。
2. (オプション) どこで今回の勉強会の開催を知りましたか?
3. (オプション) 何について聞きたい/参加者と話をしたいですか?

この課題に対して提出いただいた内容は以下です。

## 1.1 野島

1. Q.hack time に何をしますか?  
A. RC デバッグ、Debian unstable 上の評価、バグレボ書きなど。RC は <https://bugs.debian.org/release-critical/> 参照。
2. (オプション)Q. 何について聞きたい/参加者と話をしたいですか?  
A. 自由ソフトウェアと自由な環境と自由な技術についての熱い思い! 自由ソフトウェアと自由な環境と技術で社会へ何をアピールした方があなたにとって Happy か、そして、欲望を、いたずら心を満たせるのか。

## 1.2 alohaug

1. Q.hack time に何をしますか?  
A. debian livedcd for gnuk key generation
2. (オプション)Q. どこで今回の勉強会の開催を知りましたか?  
A. その他
3. (オプション)Q. 何について聞きたい/参加者と話をしたいですか?  
A. gnuk や秘密鍵保管オープンハードの昨今

## 1.3 henrich

1. Q.hack time に何をしますか?  
A. 多分原稿書いてます…。
2. (オプション)Q. どこで今回の勉強会の開催を知りましたか?

- A. 友達や知り合いから直接
3. (オプション)Q. 何について聞きたい/参加者と話をしたいですか?  
A. 皆が Debian に関して知りたいことを把握したいです

## 1.4 koedoyoshida

1. Q.hack time に何をしますか?  
A. まったりメンテナンスなど。
2. (オプション)Q. どこで今回の勉強会の開催を知りましたか?  
A. Debian JP のメーリングリスト

## 1.5 dictoss

1. Q.hack time に何をしますか?  
A. Debian GNU/kFreeBSD をネタにしたプレゼンテーション準備
2. (オプション)Q. どこで今回の勉強会の開催を知りましたか?  
A. Debian JP のメーリングリスト

## 1.6 yyatsuo

1. Q.hack time に何をしますか?  
A. 作業環境の整備した後放置している Debian 関連のドキュメントをなんとかする。
2. (オプション)Q. どこで今回の勉強会の開催を知りましたか?  
A. その他

## 1.7 yy-y-ja-jp

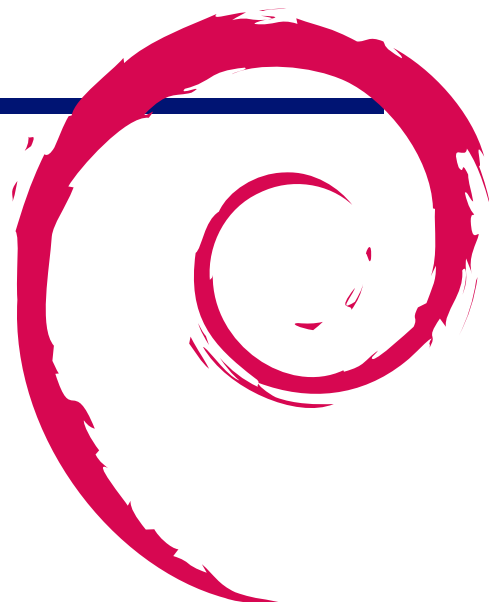
1. Q.hack time に何をしますか？  
A. DDTSS。
2. (オプション)Q. どこで今回の勉強会の開催を知りましたか？  
A. Debian JP のメーリングリスト。

## 1.8 Roger Yan

1. Q.hack time に何をしますか？  
A. 未定。
2. (オプション)Q. どこで今回の勉強会の開催を知りましたか？  
A. その他。

## 2 Debian Trivia Quiz

野島 貴英



Debian の昨今の話題についての Quiz です。

今回の出題範囲は `debian-devel-announce@lists.debian.org` や `debian-news@lists.debian.org` に投稿された内容などからです。

問題 1. 12/29 に、Debian のサービスと連携できる chromium 向けの拡張機能がリリースされました。何と連携できるのでしょうか？

- A tracker.debian.org
- B sources.debian.net
- C www.debian.or.jp

問題 2. 2014/11/11 に Raphaël Hertzog さんにより提案された DEP14 の内容は以下のどれ？

- A git パッケージングリポジトリの使い方について
- B control ファイルに Debian-Homepage フィールドを入れる件
- C debian/upstream/metadata を用意する件

問題 3. 2015/1/8 Marvel 社より Debian ARM 用パッケージビルドマシンの寄付がありました。さて何台の提供があったでしょう？

- A 7 台
- B 8 台
- C 10 台

問題 4. 2015/1/10 に wheezy がアップデートされました。さてバージョンは次のどれ。

- A 7.6
- B 7.7
- C 7.8

問題 5. 2015/1/1 にある長さ未満の Debian keyring の消去が完了したそうです。ある長さとは？

- A 2048
- B 4096
- C 8192

問題 6. 2014/12/6 に `cdn.debian.net` のレコードの示す先を、とある FQDN の CNAME としたい旨の問い合わせが `debian-devel` ML にありました。どこに向ける？

- A ftp.debian.or.jp
- B ftp.jp.debian.org
- C http.debian.net

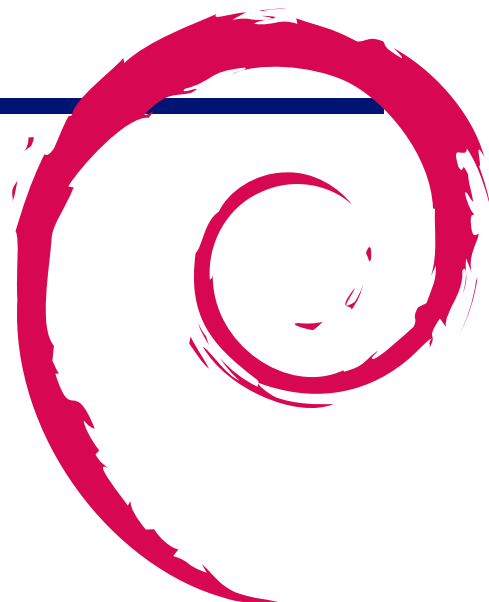
問題 7. 2015/1/9 に `debian-devel` ML にて、一部パッケージの `description` が長すぎる件についての議論がありました。最も長い `description` の行を持つパッケージは次のどれ？

- A firmware-linux-nonfree
- B texlive-latex-extra
- C irssi-scripts

## 3 最近の Debian 関連のミーティング報告

野島 貴英

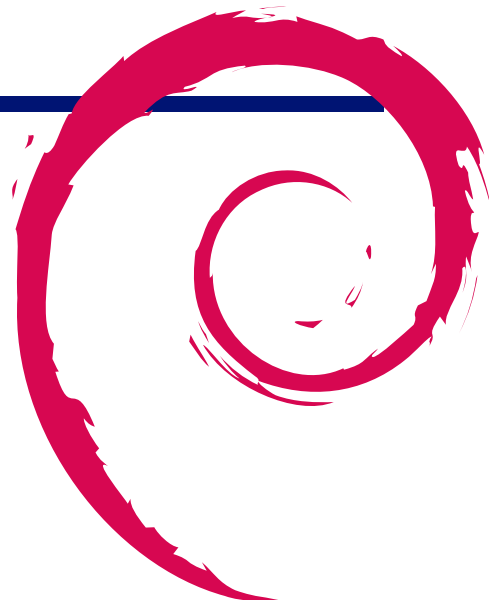
---



### 3.1 第 121 回東京エリア Debian 勉強会

- 場所はスクウェア・エニックスさんのセミナールームをお借りしての開催でした。
- 参加者は 9 名でした。
- セミナ内容は
  - @kenhys さんにより「Debian と Fedora でパッケージをリリースするまでの話」
  - 野島さんにより、「Debian から見た Linux Mint」でした。
- 残りの時間で hack time を行い、成果発表をしました。
- 宴会の代わりに、「大戸屋ごはん処 新宿イーストサイドスクエア」で夕食会をやりました。

セミナですが、@kenhys さんにより、groonga についての概要と Debian/Fedora へ公式パッケージをリリースするまでが語られました。現在の Debian/Fedora の公式パッケージ化に関して、実際のところが語られました。実際に公式パッケージにするまでの諸々について語られる機会は、他ではあまり見ない件なので、非常に貴重な発表だったと思います。また、野島さんにより、Linux Mint について、Debian との比較について語られました。



## 4 Emacs 関連パッケージの Debian パッケージ作成

henrich

### 4.1 はじめに

Debian で elisp パッケージをお作法に従って作る方法について語ってみます。  
emacs 関連パッケージの Debian パッケージ化の手続きを簡単にいうと、

Step 1. emacsen-common パッケージを入れる

Step 2. /usr/share/doc/emacsen-common/debian-emacs-policy.gz を読む

Step 3. ポリシーに従って作る

以上となります。

…というのも突き放しすぎなので、step by step で作り方を説明してみます。

### 4.2 step by step での作り方

まず、ソースを展開しておいて、dh-make パッケージを入れて dh\_make コマンドでパッケージを作ります。パッケージは dh-make、コマンドは dh\_make と「-」と「\_」の違いがあるので注意。この際、\*.el なファイルがあるとテンプレートファイルをコピーしてパッケージ名に合わせて修正してくれます（あるいは `-with-emacs` オプションを使います）。以前は特に .el なファイルがなくてもテンプレートファイルがコピーされていたのですが、毎度消すのが鬱陶しかったのでパッチを送って修正してもらいました (\*1)

```
$ sudo apt-get install dh-make
$ dh_make --createorig

Type of package: single binary, indep binary, multiple binary, library, kernel module, kernel patch?
[s/i/m/l/k/n] s

Maintainer name : Hideki Yamane
Email-Address   : henrich@XXXXXX
Date            : Sun, 04 Jan 2015 12:26:09 +0900
Package Name    : ag-el
Version         : 0.44
License         : blank
Type of Package : Single
Hit <enter> to confirm:
Currently there is no top level Makefile. This may require additional tuning.
Done. Please edit the files in the debian/ subdirectory now. You should also
check that the ag.el Makefiles install into $DESTDIR and not in / .
```

\*1 <http://bugs.debian.org/696793>

```
$ ls debian/
README.Debian      changelog  emacsen-install.ex  manpage.sgml.ex  preinst.ex
README.source      compat    emacsen-remove.ex  manpage.xml.ex   prerm.ex
ag.el.cron.d.ex    control   emacsen-startup.ex  menu.ex           rules
ag.el.default.ex   copyright init.d.ex           postinst.ex       source
ag.el.doc-base.EX docs      manpage.1.ex        postrm.ex         watch.ex
```

emacsen-\*.ex ファイルがあることがわかります。これをリネームして.ex ポストフィックスを外してやりましょう。余分なファイルも消しておきます。

```
$ ls
changelog  control  docs      emacsen-remove  rules  watch
compat    copyright emacsen-install emacsen-startup source
```

スッキリしました。emacsen\* ファイルの中身は特にいじる必要はありません。次に debian/control を修正します。

```
$ vi debian/control
-----debian/control ファイルここから-----
Package: ag-el
Architecture: any
Depends: ${shlibs:Depends}, ${misc:Depends}
Description: <insert up to 60 chars description>
<insert long description, indented with spaces>

Package: ag-el
Architecture: all
Depends: ${shlibs:Depends}, ${misc:Depends}, emacsen-common (>= 2.0.8), emacs,
silversearcher-ag,
Description: Emacs frontend to ag
The Silver Searcher (a.k.a. ag) is very fast grep-like program.
It is faster and has an attractive features than grep.
ag.el is simple ag frontend for Emacs, loosely based on ack-and-half.el.
-----debian/control ファイルここまで-----
```

Architecture: は elisp スクリプトなので all に変えます。ポイントは Depends 行の emacsen-common (>= 2.0.8), emacs の 2 つです。

emacsen-common (>= 2.0.8) の必要性は Emacs Policy に書いてあります。これと併せて debian/emacs-compat ファイルを置きます (中身は単に「0」とだけする)。このファイルは dh\_installemacsen が処理してくれます。man dh\_installemacsen とすると記載があります。

```
$ man dh_installemacsen
... 中略...
FILES
    debian/package.emacsen-compat
        Installed into usr/lib/emacsen-common/packages/compat/package in the package build directory.
... 中略...
```

emacs はメタパッケージになっており、Emacs24 などの特定バージョンに依存しています。Depends: emacs24 などとすると、Emacs25 が出た時に再度 Depends 行を書き換える必要が発生しますが、このメタパッケージを指定すればパッケージ側での都度変更は不要です。

```
$ apt-cache show emacs
Package: emacs
Source: emacs-defaults
Version: 46.1
Installed-Size: 25
Maintainer: Rob Browning <rlb@defaultvalue.org>
Architecture: all
Depends: emacs24 | emacs24-lucid | emacs24-nox
Description-jp: GNU Emacs エディタ (メタパッケージ)
GNU Emacs は、拡張可能で自己説明的なテキストエディタです。
これは常に推奨の最新 Emacs リリースに依存するメタパッケージです。
Description-md5: 21fb7da111336097a2378959f6d6e6a8
Tag: devel::editor, role::dummy, role::metapackage, suite::emacs, suite::gnu,
use::editing
Section: editors
Priority: optional
Filename: pool/main/e/emacs-defaults/emacs_46.1_all.deb
Size: 1634
MD5sum: 1f115942065ac452467e02377368ee22
SHA1: dbb1343a3d24f60e5038994e3528dd7486e40943
SHA256: c1fad54e790d69b83f32f2612963baba3ea8091ff3ca72c960c7312096223e3a
```

その他にも Xemacs を含む emacsen というパッケージでの指定もありますが、これは仮想パッケージなので単独



で指定はできません。指定するとしたら”Depends: emacs — emacsen” のようにします。

…とここまで書いて「dh-make でフォローしてくれればいいんじゃないの？」と気づきました。では、dh-make パッケージへのパッチを作りましょう…できましたので、BTS を…しました。将来的にはこの部分は「へーそんなのもあるんだー」的になるはず。あ、reportbug で複数ファイルを添付するには 1 つずつ -A で指定が必要なようです。

```
$ debcheckout dh-make
$ cd dh-make
$ git checkout -b support-modern-emacs-policy
... いろいろ変更...
$ git format-patch master
$ ls
0001-add-emacsen-compat-for-modern-Emacs-lisp-package.patch 0002-add-debian-control-file-for-Emacs-add-on.patch
debian dh_make dh_make.1 lib
$ reportbug -A 0001-add-emacsen-compat-for-modern-Emacs-lisp-package.patch \
-A 0002-add-debian-control-file-for-Emacs-add-on.patch dh-make
```

これで Bug#774545 として登録されました。

しかし、ここで気になることが。dh-make での debian/control ファイルはどれも種類に合わせてコピーされるようです。これをそのまま適用すると、\*.el ファイルがある場合に否応なしに Emacs add-on パッケージ用の control ファイルになるわけで、複数のバイナリパッケージを生成するような場合には嬉しくありません。

多分、大本の control ファイルがあつて、Emacs 用のが追加される形のほうが嬉しい…のでは？という気がします。…しかし、その場合の「Section: lisp」指定をどうするのか…

```
Source: #PACKAGE#
Section: lisp
Priority: optional
Maintainer: #USERNAME# <#EMAIL#>
Build-Depends: #BUILD_DEPS#
Standards-Version: #POLICY#
Homepage: <insert the upstream URL, if relevant>
#Vcs-Git: git://anonscm.debian.org/collab-maint/#PACKAGE#.git
#Vcs-Browser: http://anonscm.debian.org/cgit/collab-maint/#PACKAGE#.git/

Package: #PACKAGE#
Architecture: all
Depends: ${misc:Depends}, emacsen-common (>= 2.0.8), emacs | emacsen,
Description: <insert up to 60 chars description>
<insert long description, indented with spaces>
```

これ、パッケージののところだけを「追記」する形だといいいのですかね？

```
Package: #PACKAGE#
Section: lisp
Architecture: all
Depends: ${misc:Depends}, emacsen-common (>= 2.0.8), emacs | emacsen,
Description: <insert up to 60 chars description>
<insert long description, indented with spaces>
```

そうすればソースパッケージの雛形も 1 箇所ですべてまとめれば済んで、いい感じになるんじゃないか？とか思い始めました。

```
Source: #PACKAGE#
Section: UNKNOWN
Priority: optional
Maintainer: #USERNAME# <#EMAIL#>
Build-Depends: #BUILD_DEPS#
Standards-Version: #POLICY#
Homepage: <insert the upstream URL, if relevant>
#Vcs-Git: git://anonscm.debian.org/collab-maint/#PACKAGE#.git
#Vcs-Browser: http://anonscm.debian.org/cgit/collab-maint/#PACKAGE#.git/
```

でもそれは別ハックですね…。

あとは rebuild して lintian の warning などをちょこちょこ直せば完成です。

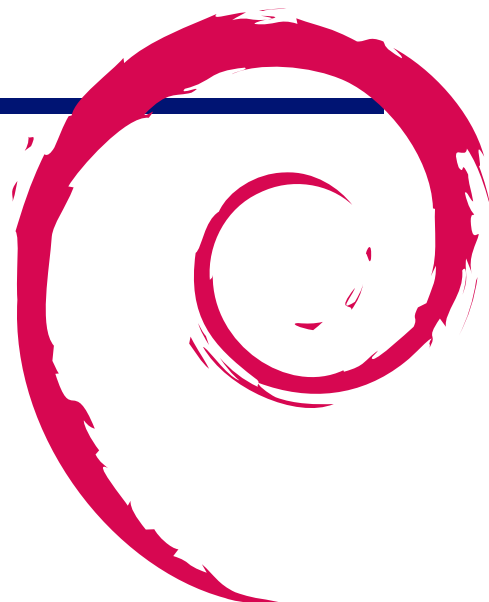
```

$ dpkg --contents /var/cache/pbuilder/result/ag-el_0.44-1_all.deb
drwxr-xr-x root/root          0 2015-01-04 13:35 ./
drwxr-xr-x root/root          0 2015-01-04 13:35 ./usr/
drwxr-xr-x root/root          0 2015-01-04 13:35 ./usr/share/
drwxr-xr-x root/root          0 2015-01-04 13:35 ./usr/share/doc/
drwxr-xr-x root/root          0 2015-01-04 13:35 ./usr/share/doc/ag-el/
-rw-r--r-- root/root        175 2015-01-04 13:10 ./usr/share/doc/ag-el/changelog.Debian.gz
-rw-r--r-- root/root       1089 2015-01-04 13:35 ./usr/share/doc/ag-el/copyright
-rw-r--r-- root/root       5279 2014-08-05 05:58 ./usr/share/doc/ag-el/README.md.gz
drwxr-xr-x root/root          0 2015-01-04 13:35 ./usr/lib/
drwxr-xr-x root/root          0 2015-01-04 13:35 ./usr/lib/emacsen-common/
drwxr-xr-x root/root          0 2015-01-04 13:35 ./usr/lib/emacsen-common/packages/
drwxr-xr-x root/root          0 2015-01-04 13:35 ./usr/lib/emacsen-common/packages/remove/
-rwxr-xr-x root/root         465 2015-01-04 13:35 ./usr/lib/emacsen-common/packages/remove/ag-el
drwxr-xr-x root/root          0 2015-01-04 13:35 ./usr/lib/emacsen-common/packages/install/
-rwxr-xr-x root/root       1280 2015-01-04 13:35 ./usr/lib/emacsen-common/packages/install/ag-el
drwxr-xr-x root/root          0 2015-01-04 13:35 ./usr/lib/emacsen-common/packages/compat/
-rw-r--r-- root/root          2 2015-01-04 13:35 ./usr/lib/emacsen-common/packages/compat/ag-el
drwxr-xr-x root/root          0 2015-01-04 13:35 ./etc/
drwxr-xr-x root/root          0 2015-01-04 13:35 ./etc/emacs/
drwxr-xr-x root/root          0 2015-01-04 13:35 ./etc/emacs/site-start.d/
-rw-r--r-- root/root       1225 2015-01-04 13:35 ./etc/emacs/site-start.d/50ag-el.el

```

### 4.3 まとめ

- elisp パッケージは emacsen-common パッケージを入れて /usr/share/doc/emacsen-common/debian-emacs-policy.gz ファイルに従ってパッケージを作ろう
- お手軽にするには dh-make パッケージを使おう
- Policy を読んで tips を確認する手間を省くには dh-make パッケージをハックしてしまえばいい



## 5 会場での無線 LAN のつなぎ方

野島 貴英,Roger

### 5.1 はじめに

今回試験として、会場側でフィルタ無しのグローバル回線を用意しました。ただ、会場側のセキュリティポリシーにより、wpa-psk AES hidden SSID という方式での提供となります。

以下に Debian マシンでの接続方法を記載します。

また、自分の環境では違うやり方でつながったという方は、野島まで教えて下さい。こちらでもノウハウとして溜めていく予定です。

### 5.2 wpa\_supplicant 及び/etc/network/interfaces を利用の場合

もっとも良いマニュアルは、`/usr/share/doc/wpa_supplicant/README.Debian.gz` となります。困った場合はこちらも合わせてご参照下さい。

以下に/etc/network/interfaces の定義について会場の例を記載します。

```
$ sudo vi /etc/network/interfaces
----以下のエントリがなければ追記ここから-----
iface wlan0_debian inet dhcp
    wpa-conf /etc/wpa_supplicant/wpa_supplicant_debian.conf
----以下のエントリがなければ追記ここまで-----
$ sudo vi /etc/wpa_supplicant/wpa_supplicant_debian.conf
----以下のエントリを追記ここから-----
network={
    ssid=<<会場の SSID>>
    psk=<<会場のパスワード>>
    scan_ssid=1
}
----以下のエントリを追記ここまで-----
$ sudo chmod 600 /etc/wpa_supplicant/wpa_supplicant_debian.conf
$ sudo ifup wlan0=wlan0_debian
```

また、ハマってしまった時のデバッグ方法は、`/usr/share/doc/wpa_supplicant/README.Debian.gz` 中の”4. Troubleshooting” の章が便利です。

### 5.3 その他の無線 LAN 用パッケージを利用の場合

すみません、自分が情報を持たないため、現場で教えて下さい。



**Debian 勉強会資料**

2015 年 1 月 17 日 初版第 1 刷発行

東京エリア Debian 勉強会（編集・印刷・発行）

---