

.Debian

銀河系唯一のDebian専門誌

2017年4月22日

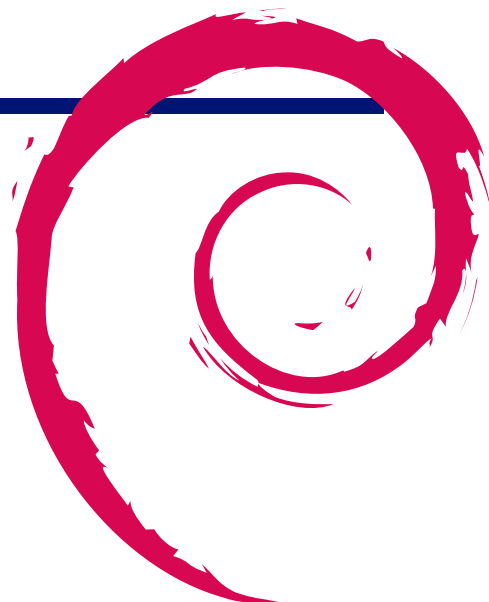
特集：パッケージカスタマイズ



会 勉 強 会 の ア ー ビ ト

目次

1	最近の Debian 関連のミーテ ィング報告	2	2.7	Roger Shimizu	3
1.1	第 148 回東京エリア Debian 勉強会	2	2.8	hatochan	3
1.2	OSC 2017 Tokyo/Spring 出 展 (第 149 回東京エリア De- bian 勉強会)	2	2.9	kenhys	3
2	事前課題	3	2.10	Marc Dequènes (Duck)	3
2.1	Charles Plessy	3	2.11	koedoyoshida	4
2.2	ftake	3	2.12	takaswie	4
2.3	NOKUBI Takatsugu	3	3	Debian Trivia Quiz	5
2.4	yy-y-ja- <i>jp</i>	3	4	Debian パッケージのカスタマ イズについて	6
2.5	nabaua	3	4.1	はじめに	6
2.6	dictoss	3	4.2	前提	6
			4.3	Debian パッケージングの部分	7
			4.4	上流のコード自体	13
			4.5	メンテナンス	17
			4.6	まとめ	17
			4.7	参考文献	17



1 最近の Debian 関連のミーティング報告

杉本 典充

1.1 第 148 回東京エリア Debian 勉強会

2017 年 2 月 11 日 (土) に第 148 回東京エリア Debian 勉強会を開催しました。会場は銀座にある朝日ネットさんをお借りして行いました。参加者は 9 名でした。

2017 年 2 月 5 日に Debian 9 stretch の Full Freeze 宣言がアナウンスされました^{*1}。今回の勉強会では、リリースクリティカルバグを解決するべく作業を行う「バグ・スカッシュ・パーティー」を行いました。

Debian Developer である岩松さんからリリースクリティカルバグの調べ方、patch の投稿作法、BTS へのタグの効果的なつけ方を参加者へ共有しました。その後、リリースクリティカルバグの中で難易度の高いものと低いものを選別し、参加者は解決できるものからバグ解決に臨みました。

バグ・スカッシュ・パーティーの結果は、以下の Debian wiki にまとめています。

<https://wiki.debian.org/BSP/2017/02/jp/Tokyo>

1.2 OSC 2017 Tokyo/Spring 出展 (第 149 回東京エリア Debian 勉強会)

2017 年 3 月 11 日 (土) に開催された OSC 2017 Tokyo/Spring へ、東京エリア Debian 勉強会/DebianJP として出展しました。

展示ブースでは Debian GNU/Linux をインストールした PC 及び ARM と FPGA の両方を搭載したリファレンスボードの展示、Debian Project と東京エリア Debian 勉強会の広報活動を行いました。また、OSC 参加者と交流を深めました。

セミナーは「第 149 回東京エリア Debian 勉強会」として開催し、「Debian updates」という表題で岩松さんが発表しました。また、セミナーに参加した 25 名の方々から疑問や質問に回答しました。

^{*1} <https://lists.debian.org/debian-devel-announce/2017/02/msg00001.html>

2 事前課題

杉本 典充



今回の事前課題は以下です:

1. Hack Time は何をしますか。
2. 普段使っている Debian パッケージのうち、カスタマイズ (コンパイルオプション変更や独自修正など) しているものがあれば、そのパッケージ名とカスタマイズ内容を書いてください。

この課題に対して提出いただいた内容は以下です。

2.1 Charles Plessy

1. Fix R packages
2. 無し

2.2 ftake

1. openSUSE.Asia Summit の開催プロポーザルを書いています。
2. 新しい Boost が欲しいことがあり、unstable から引っ張ってきてビルドしていたことがありました

2.3 NOKUBI Takatsugu

1. BTS 対応
2. 最近はやってません

2.4 yy-y-ja-jp

1. DDTSS
2. 最近では firefox の拡張パッケージを新しいバージョンに入れ替えるぐらいです

2.5 nabaua

1. その日の考えます。
2. カスタマイズはないです。

2.6 dictoss

1. RFA になっている snmptt の upstream およびパッケージの中身の調査
2. backports で問題ないため、既存パッケージのカスタマイズはしていない。

2.7 Roger Shimizu

1. パッケージメンテナとか、バグ潰しなどするつもり
2. 現在特にありません

2.8 hatochan

1. debian インストールまたは live-help をごによごによ
2. firmware 関連のパッケージを試行錯誤

2.9 kenhys

1. パッケージのメンテナンス
2. 独自にメンテするのが面倒なので基本はしない 必要ならフィードバックしてパッケージ側に反映してもらう

2.10 Marc Dequènes (Duck)

1. BSP とかメンター作業など
2. backport とか pending fix などのためにカスタマイズをよくします。

2.11 koedoyoshida

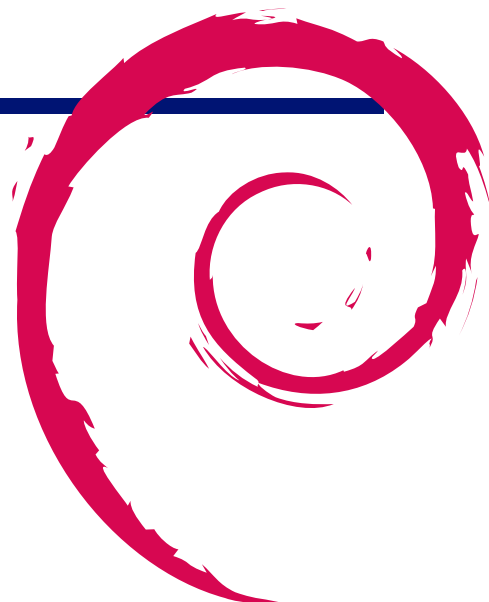
1. Debian 関係作業、PyConJP 関係作業
2. 最近は無し

2.12 takaswie

1. libhinawa の新リリースの準備作業
2. network-manager。
dpkg-divert で/usr/lib/NetworkManager/conf.d/
10-globally-managed-devices.conf を退避。
[https://bugs.launchpad.net/ubuntu/+source/
network-manager/+bug/1638842](https://bugs.launchpad.net/ubuntu/+source/network-manager/+bug/1638842) の問題のため。

3 Debian Trivia Quiz

杉本 典充



Debian の昨今の話題についての Quiz です。

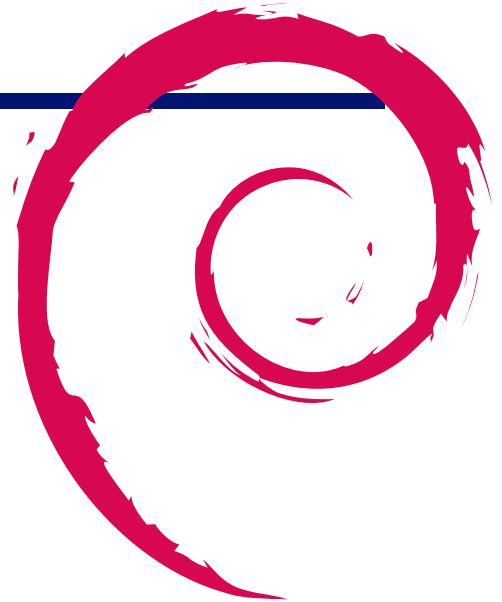
今回の出題範囲は `debian-devel-announce@lists.debian.org` や `debian-news@lists.debian.org` などに投稿された内容からです。

問題 1. 2017 年の Debian Project Leader(=DPL) 選挙が行われ、投票結果が発表されました。DPL に選ばれたのは誰でしょうか。

- A Mehdi Dogguy
- B Nobuhiro Iwamatsu
- C Chris Lamb

4 Debian パッケージのカスタマイズについて

yy-y-ja-jp



4.1 はじめに

Debian パッケージをカスタマイズしたいけどやり方が (特にどこに書いてあるのか) 分からないという話があったので、まとめてみました。

まずパッケージングの前提を見ていきます。

次に、カスタマイズしたい場所は大きく 2 つあるでしょう。

- Debian パッケージングの部分
- 上流のコード自体

また、これらカスタマイズした内容をその後メンテナンスしていく必要があります。それぞれ見ていきましょう。

4.2 前提

パッケージングについての概要は “パッケージングチュートリアル” にあるので `apt install packaging-tutorial` を実行してから `/usr/share/doc/packaging-tutorial/` の中を見てみましょう*2。

カスタマイズするにはパッケージのソースコードを手元に用意する必要があります。 `apt-get source` や `dget(1)` (`devscripts` パッケージにあります) でダウンロードできます。メンテナがパッケージング作業用の `git` レポジトリなどを公開している場合 (`Vcs-*` フィールド) は、 `debcheckout(1)` (`devscripts` パッケージ) も使うとそのレポジトリを `clone` したりしてくれるようです。

そして、Debian パッケージは “Debian ポリシー” (`debian-policy` パッケージ) に従って作られています。Debian ポリシーは、Debian がうまく動くように定められたルールです。カスタマイズしたパッケージもポリシーに従ったほうがよいでしょう。本当はポリシー全文を読むとよいでしょう、と言いたいところですが長文です。また、Debian ポリシー以外にも従っておいたほうがよい慣習が “Debian 開発者リファレンス” (`developers-reference-ja` パッケージ)*3 などいくつか書かれています。

`debhelper` などのパッケージングヘルパーはポリシーにできるだけ自動的に従ってパッケージが作られるように仕組んであります。なので普通にパッケージングヘルパーを使ってパッケージング作業をしていればある程度はポリシーに勝手に従っているはずですが。

また、 `lintian(1)` というツール (`lintian` パッケージ) が Debian ポリシーに違反していないか、慣習に沿っているかチェックしてくれます。 `lintian` はインストール済みならパッケージングコマンド `debuild` の最後で自動的に呼ばれ

*2 <https://www.debian.org/doc/manuals/packaging-tutorial/packaging-tutorial.ja.pdf> にもあります。

*3 <https://www.debian.org/doc/manuals/developers-reference/index.ja.html> にもあります。

ます。実際のところいろんな慣習がいろんな場所に書いてあってわからないことも多いので、まずは作ったパッケージを rebuild してみて、lintian の結果を見て、エラーや警告が出たら詳細説明を読んで修正しましょう。

特に、慣習の 1 つにはバージョン番号があります。開発者レファレンスによると、パッケージメンテナ以外がパッケージを変更して Debian 本体にアップロードするとき (Non-Maintainer Upload, NMU) には、変更内容を debian/changelog に書き、特別なバージョン番号にしなければなりません。常にメンテナのアップロードするバージョンが優先されるように、次にメンテナがアップロードするであろうバージョン番号よりも低くしているからです。本当に NMU するわけではないにせよ重要なので、これについては以降も見ていきます。

4.3 Debian パッケージングの部分

これについては

- コンパイルオプションなどを変更したい
- 依存関係やパッケージ構成などを変更したい
- 不安定版 (unstable) にあるバージョンを安定版 (stable) などにバックポートしたい

などがあるでしょう。

4.3.1 コンパイルオプション

パッケージングでは debian/rules ファイルからコンパイルが実行されます。コンパイルオプションを変更するときは、これを変更するだけのことが多いです (正確には、一通り変更して、その変更内容を dch(1) コマンドで debian/changelog ファイルに新しいバージョン番号で記入したら完成です。以降も同様です)。

ここでは例として hello というパッケージを見てみます。

```
$ apt-get source hello
パッケージリストを読み込んでいます... 完了
733 kB のソースアーカイブを取得する必要があります。
取得:1 http://ftp.jp.debian.org/debian sid/main hello 2.10-1 (dsc) [1,323 B]
取得:2 http://ftp.jp.debian.org/debian sid/main hello 2.10-1 (tar) [726 kB]
取得:3 http://ftp.jp.debian.org/debian sid/main hello 2.10-1 (diff) [6,072 B]
(snip)
$ cd hello-2.10/
```

```
$ editor debian/rules
```

```
#!/usr/bin/make -f
%:
    dh $@

override_dh_auto_clean:
    [ ! -f Makefile ] || $(MAKE) distclean

override_dh_installdocs:
    dh_installdocs NEWS
```

この hello にはあまり面白そうなオプションが見当たらないのですが、とりあえず ./configure に --disable-nls を付けてみます。さて、どこに書けばいいかわからないのでとりあえず rebuild してみます。

```
$ rebuild -us -uc
dpkg-buildpackage -rfakeroot -us -uc
dpkg-buildpackage: info: source package hello
dpkg-buildpackage: info: source version 2.10-1
dpkg-buildpackage: info: source distribution unstable
(snip)
debian/rules build
dh build
  dh_testdir
  dh_update_autotools_config
  dh_auto_configure
  ./configure --build=x86_64-linux-gnu --prefix=/usr --includedir=${prefix}/include --mandir=${prefix}/share/man --infodir=${prefix}/share/info
configure: WARNING: unrecognized options: --disable-maintainer-mode
(snip)
```


どうやら `dh_auto_configure` の man ページを読むと、これにオプションを渡せばいいようなのでそうしてみます。

```
#!/usr/bin/make -f
%:
    dh $@

override_dh_auto_clean:
    [ ! -f Makefile ] || $(MAKE) distclean

override_dh_installdocs:
    dh_installdocs NEWS

override_dh_auto_configure:
    dh_auto_configure -- --disable-nls
```

`debian/changelog` を書きます。 `dch` を実行すると^{*4}、メンテナではないため自動的に `--nmv` モードになります。ディストリビューションが `UNRELEASED` に仮で書かれます。

```
$ dch
$ head -n 10 debian/changelog
hello (2.10-1.1) UNRELEASED; urgency=medium

 * Non-maintainer upload.
 * debian/rules: Use --disable-nls.

-- YOSHINO Yoshihito <yy.y.ja.jp@gmail.com> Sat, 22 Apr 2017 08:58:38 +0900

hello (2.10-1) unstable; urgency=low

 * New upstream release.
```

とりあえず `debuild` してみます。最後に `lintian` が実行されてるので見ましょう。

```
$ debuild -us -uc
dpkg-buildpackage -rfakeroot -us -uc
dpkg-buildpackage: info: source package hello
dpkg-buildpackage: info: source version 2.10-1.1
dpkg-buildpackage: info: source distribution unstable
(snip)
dh_md5sums
dh_builddeb
dpkg-deb: building package 'hello-dbgsym' in '../hello-dbgsym_2.10-1.1_amd64.deb'.
dpkg-deb: building package 'hello' in '../hello_2.10-1.1_amd64.deb'.
dpkg-genbuildinfo
dpkg-genchanges >../hello_2.10-1.1_amd64.changes
dpkg-genchanges: info: not including original source code in upload
dpkg-source --after-build hello-2.10
dpkg-buildpackage: info: binary and diff upload (original source NOT included)
Now running lintian...
W: hello source: ancient-standards-version 3.9.6 (current is 3.9.8)
Finished running lintian.
```

`lintian` の各行の詳細説明は `lintian-info(1)` で見られます^{*5}。

```
$ lintian-info --tags ancient-standards-version
W: ancient-standards-version
N:
N: The source package refers to a Standards-Version that has been
N: obsolete for more than two years. Please update your package to latest
N: Policy and set this control field appropriately.
N:
N: If the package is already compliant with the current standards, you
N: don't have to re-upload the package just to adjust the
N: Standards-Version control field. However, please remember to update
N: this field next time you upload the package.
N:
N: See /usr/share/doc/debian-policy/upgrading-checklist.txt.gz in the
N: debian-policy package for a summary of changes in newer versions of
N: Policy.
N:
N: Refer to https://www.debian.org/doc/debian-policy/upgrading-checklist
N: for details.
N:
N: Severity: normal, Certainty: certain
N:
N: Check: standards-version, Type: source
N:
```

^{*4} 実行前に `DEBFULLNAME`, `DEBEMAIL` 環境変数を設定してください。詳しくは“新メンテナガイド”(後述)にあります。

^{*5} <https://lintian.debian.org/tags/ancient-standards-version.html> でも見られますが、問題のある全パッケージが載っているので重いです。

なお、この例ではカスタマイズする前から `ancient-standards-version` が出ているようなので、修正せずそのまましておきます。メンテナが次にリリースしたときに直すでしょうし、それに対して自分のカスタマイズをしたときに、自分のカスタマイズ内容だけを変更すればよいからです。

カスタマイズが完了したらディストリビューションを確定させます。以下を実行してエディタで保存します。

```
$ dch -r
$ head -n 10 debian/changelog
hello (2.10-1.1) unstable; urgency=medium

* Non-maintainer upload.
* debian/rules: Use --disable-nls.

-- YOSHINO Yoshihito <yy.y.ja.jp@gmail.com> Sat, 22 Apr 2017 09:00:30 +0900

hello (2.10-1) unstable; urgency=low

* New upstream release.
```

debuild すれば完成です。

```
$ debuild -us -uc
dpkg-buildpackage -rfakeroot -us -uc
dpkg-buildpackage: info: source package hello
dpkg-buildpackage: info: source version 2.10-1.1
dpkg-buildpackage: info: source distribution unstable
(snip)
dh_md5sums
dh_builddeb
dpkg-deb: building package 'hello-dbgSYM' in './hello-dbgSYM_2.10-1.1_amd64.deb'.
dpkg-deb: building package 'hello' in './hello_2.10-1.1_amd64.deb'.
dpkg-genbuildinfo
dpkg-genchanges >./hello_2.10-1.1_amd64.changes
dpkg-genchanges: info: not including original source code in upload
dpkg-source --after-build hello-2.10
dpkg-buildpackage: info: binary and diff upload (original source NOT included)
Now running lintian...
W: hello source: ancient-standards-version 3.9.6 (current is 3.9.8)
Finished running lintian.
```

さて、ここまで (`dch` と引数なしで起動することで) NMU するパッケージング作業をしてきました。Debian 本体にアップロードするのではなく個人的に使ったりするためのカスタマイズであれば、NMU ではなく “ローカルアップロード” にしたほうがよいでしょう。ローカルアップロードには名前を付けます。ここでは名前を `ysn` にしてみます。

```
$ dch --local ysn
```

```
$ head -n 10 debian/changelog
hello (2.10-1ysn1) UNRELEASED; urgency=medium

* debian/rules: Use --disable-nls.

-- YOSHINO Yoshihito <yy.y.ja.jp@gmail.com> Sat, 22 Apr 2017 08:44:10 +0900

hello (2.10-1) unstable; urgency=low

* New upstream release.
* debian/patches: Drop 01-fix-i18n-of-default-message, no longer needed.
(snip)
```

debuild してみます。

```
$ debuild -us -uc
dpkg-buildpackage -rfakeroot -us -uc
dpkg-buildpackage: info: source package hello
dpkg-buildpackage: info: source version 2.10-1ysn1
dpkg-buildpackage: info: source distribution UNRELEASED
(snip)
dh_md5sums
dh_builddeb
dpkg-deb: building package 'hello-dbgSYM' in './hello-dbgSYM_2.10-1ysn1_amd64.deb'.
dpkg-deb: building package 'hello' in './hello_2.10-1ysn1_amd64.deb'.
dpkg-genbuildinfo
dpkg-genchanges >./hello_2.10-1ysn1_amd64.changes
dpkg-genchanges: info: not including original source code in upload
dpkg-source --after-build hello-2.10
dpkg-buildpackage: info: binary and diff upload (original source NOT included)
Now running lintian...
W: hello source: changelog-should-mention-nmu
W: hello source: source-nmu-has-incorrect-version-number 2.10-1ysn1
W: hello source: ancient-standards-version 3.9.6 (current is 3.9.8)
Finished running lintian.
```

lintian が警告しているようなので見てみます。

```
$ lintian-info --tags changelog-should-mention-nmu source-nmu-has-incorrect-version-number
W: changelog-should-mention-nmu
N:
N: When you NMU a package, that fact should be mentioned on the first
N: line in the changelog entry. Use the words "NMU" or "Non-maintainer
N: upload" (case insensitive).
N:
N: Maybe you didn't intend this upload to be a NMU, in that case, please
N: double-check that the most recent entry in the changelog is
N: byte-for-byte identical to the maintainer or one of the uploaders. If
N: this is a local package (not intended for Debian), you can suppress
N: this warning by putting "local" in the version number or "local
N: package" on the first line of the changelog entry.
N:
N: Refer to Debian Developer's Reference section 5.11.3 (Using the
N: DELAYED/ queue) for details.
N:
N: Severity: normal, Certainty: certain
N:
N: Check: nmu, Type: source
N:
```

```
W: source-nmu-has-incorrect-version-number
N:
N: A source NMU should have a Debian revision of "-x.x" (or "+nmux" for a
N: native package). This is to prevent stealing version numbers from the
N: maintainer.
N:
N: Maybe you didn't intend this upload to be a NMU, in that case, please
N: double-check that the most recent entry in the changelog is
N: byte-for-byte identical to the maintainer or one of the uploaders. If
N: this is a local package (not intended for Debian), you can suppress
N: this warning by putting "local" in the version number or "local
N: package" on the first line of the changelog entry.
N:
N: Refer to Debian Developer's Reference section 5.11.2 (NMUs and
N: debian/changelog) for details.
N:
N: Severity: normal, Certainty: certain
N:
N: Check: nmu, Type: source
N:
```

Debian にアップロードする目的のパッケージではないので、debian/changelog を修正します。

```
$ dch
$ head -n 10 debian/changelog
hello (2.10-1ysn1) UNRELEASED; urgency=medium

 * Local package.
 * debian/rules: Use --disable-nls.

-- YOSHINO Yoshihito <yy.y.ja.jp@gmail.com> Sat, 22 Apr 2017 08:44:10 +0900

hello (2.10-1) unstable; urgency=low

 * New upstream release.
```

ここでは Local package と書くことにしましたが、dch --local local でもいいでしょう。debuild します。

```
$ debuild -us -uc
dpkg-buildpackage -rfakeroot -us -uc
dpkg-buildpackage: info: source package hello
dpkg-buildpackage: info: source version 2.10-1ysn1
dpkg-buildpackage: info: source distribution UNRELEASED
(snip)
dh_md5sums
dh_builddeb
dpkg-deb: building package 'hello-dbgSYM' in './hello-dbgSYM_2.10-1ysn1_amd64.deb'.
dpkg-deb: building package 'hello' in './hello_2.10-1ysn1_amd64.deb'.
dpkg-genbuildinfo
dpkg-genchanges > ./hello_2.10-1ysn1_amd64.changes
dpkg-genchanges: info: not including original source code in upload
dpkg-source --after-build hello-2.10
dpkg-buildpackage: info: binary and diff upload (original source NOT included)
Now running lintian...
W: hello source: ancient-standards-version 3.9.6 (current is 3.9.8)
Finished running lintian.
```

問題なさそうなので dch -r して debuild すれば完成です。

4.3.2 依存関係・パッケージ構成

依存関係やパッケージ構成は debian/control に書かれているので、変更するならこれを変更するだけです。

4.3.3 バックポート

バックポートするには、基本的には dch --bpo するだけで、他の場所を変更してはいけません。バックポート先のライブラリのバージョンが古いときは debian/control の依存関係を変更する必要があることがあります。

jessie にバックポートしてみます。まず新しいバージョンのソースパッケージを手元にダウンロードします。

```
$ apt-get source -d hello
パッケージリストを読み込んでいます... 完了
733 kB のソースアーカイブを取得する必要があります。
取得:1 http://ftp.jp.debian.org/debian sid/main hello 2.10-1 (dsc) [1,323 B]
取得:2 http://ftp.jp.debian.org/debian sid/main hello 2.10-1 (tar) [726 kB]
取得:3 http://ftp.jp.debian.org/debian sid/main hello 2.10-1 (diff) [6,072 B]
733 kB を 1 秒 で取得しました (634 kB/s)
ダウンロードオンリーモードでパッケージのダウンロードが完了しました
```

このソースパッケージをバックポート先 (jessie) 環境に持っていきます。

```
$ scp -p hello_2.10.orig.tar.gz hello_2.10-1.d* jessie:~/
(snip)
```

jessie 環境で持ってきたソースパッケージを展開して作業します。

```
jessie$ dpkg-source -x hello_2.10-1.dsc
dpkg-source: info: extracting hello in hello-2.10
dpkg-source: info: unpacking hello_2.10.orig.tar.gz
dpkg-source: info: unpacking hello_2.10-1.debian.tar.xz
jessie$ cd hello-2.10/
jessie$ dch --bpo
```

* だけの行が自動的に追加されますが、今回は特に依存関係は変えてないので削除して保存します。

```
jessie$ head -n 10 debian/changelog
hello (2.10-1~bpo8+1) jessie-backports; urgency=medium

 * Rebuild for jessie-backports.

-- YOSHINO Yoshihito <yy.y.ja.jp@gmail.com> Sat, 22 Apr 2017 17:05:44 +0900

hello (2.10-1) unstable; urgency=low

 * New upstream release.
 * debian/patches: Drop 01-fix-i18n-of-default-message, no longer needed.
```

debuild してみます。

```
jessie$ debuild -us -uc
dpkg-buildpackage -rfakeroot -D -us -uc
dpkg-buildpackage: source package hello
dpkg-buildpackage: source version 2.10-1~bpo8+1
dpkg-buildpackage: source distribution jessie-backports
(snip)
Now running lintian...
E: hello changes: backports-changes-missing
W: hello source: changelog-should-mention-nmu
W: hello source: source-nmu-has-incorrect-version-number 2.10-1~bpo8+1
Finished running lintian.
```

lintian エラーが出ています。

```
jessie$ lintian-info --tags backports-changes-missing
E: backports-changes-missing
N:
N:   The changes file only has changelog entries from a single version. For
N:   backports, all changes since (old)stable or the previous backport
N:   should be listed (usually by adding the '-v' option to the build).
N:
N:   Refer to http://backports.debian.org/Contribute/ for details.
N:
N:   Severity: serious, Certainty: possible
N:
N:   Check: changes-file, Type: changes
N:
```

jessie の最新版は 2.9-2+deb8u1 のようです。

この Debian Backports のページを見ると、debuild -v2.9-2+deb8u1 とすればよいようです。

他の lintian 警告も修正してみます。

```
jessie$ dch
jessie$ head -n 10 debian/changelog
hello (2.10-1~bpo8+1) jessie-backports; urgency=medium

* Local package.
* Rebuild for jessie-backports.

-- YOSHINO Yoshihito <yy.y.ja.jp@gmail.com> Sat, 22 Apr 2017 17:05:44 +0900

hello (2.10-1) unstable; urgency=low

* New upstream release.
```

言われたように debuild してみます。

```
jessie$ debuild -v2.9-2+deb8u1 -us -uc
dpkg-buildpackage -rfakeroot -D -us -uc -v2.9-2+deb8u1
dpkg-buildpackage: source package hello
dpkg-buildpackage: source version 2.10-1~bpo8+1
dpkg-buildpackage: source distribution jessie-backports
(snip)
dh_builddeb
dpkg-deb: './hello_2.10-1~bpo8+1_amd64.deb' にパッケージ 'hello' を構築しています。
dpkg-genchanges -v2.9-2+deb8u1 >./hello_2.10-1~bpo8+1_amd64.changes
parsechangelog/debian: warning: 'since' option specifies non-existing version
parsechangelog/debian: warning: use newest entry that is earlier than the one specified
dpkg-genchanges: not including original source code in upload
dpkg-source --after-build hello-2.10
dpkg-buildpackage: binary and diff upload (original source NOT included)
Now running lintian...
Finished running lintian.
```

なんか dpkg-genchanges の警告が出てますがとりあえず解消しました。

ですが、個人的には Debian で公式なバックポートがリリースされたら更新されてほしいのでちょっと不満です。

lintian エラーはうまく消えないのですが、少しバージョン番号を下げて使うことにしています。

```
jessie$ dch
jessie$ head -n 10 debian/changelog
hello (2.10-1~bpo8+0.1) jessie-backports; urgency=medium

* Local package.
* Rebuild for jessie-backports.

-- YOSHINO Yoshihito <yy.y.ja.jp@gmail.com> Sat, 22 Apr 2017 17:05:44 +0900

hello (2.10-1) unstable; urgency=low

* New upstream release.
```

debuild します。

```
jessie$ debuild -v2.9-2+deb8u1 -us -uc
dpkg-buildpackage -rfakeroot -D -us -uc -v2.9-2+deb8u1
dpkg-buildpackage: source package hello
dpkg-buildpackage: source version 2.10-1~bpo8+0.1
dpkg-buildpackage: source distribution jessie-backports
(snip)
dpkg-genchanges -v2.9-2+deb8u1 >./hello_2.10-1~bpo8+0.1_amd64.changes
parsechangelog/debian: warning: 'since' option specifies non-existing version
parsechangelog/debian: warning: use newest entry that is earlier than the one specified
dpkg-genchanges: not including original source code in upload
dpkg-source --after-build hello-2.10
dpkg-buildpackage: binary and diff upload (original source NOT included)
Now running lintian...
E: hello changes: backports-upload-has-incorrect-version-number 2.10-1~bpo8+0.1
Finished running lintian.
```

エラー消えませんが、あきらめます。

```
$ lintian-info --tags backports-upload-has-incorrect-version-number
E: backports-upload-has-incorrect-version-number
N:
N:   The version number doesn't comply with the standard backport version
N:   rules. It should end in ~bpoX+N, where X is the release version number
N:   of the target distribution.
N:
N:   Refer to http://backports.debian.org/Contribute/ for details.
N:
N:   Severity: serious, Certainty: certain
N:
N:   Check: changes-file, Type: changes
N:
```

lintian が示している Debian Backports のページを見ても特に書いていないので、個人的にはこのまま使うことにしています。

4.4 上流のコード自体

これについては

- コードに変更を加えたい
- 新しいバージョンなどに置き換えたい

があるでしょう。

4.4.1 コードに変更を加える

Debian パッケージのバージョンによって違いますが、debian/source/format に 3.0 (quilt) と書いてある最近のパッケージであれば “quilt” というもので上流コードに対する変更差分を管理しています*⁶。quilt(1) コマンドでも作業はできるのですが、変更を加えるだけなら dpkg-source --commit が使えます。使い方は 2013 年 2 月の Debian パッケージング道場資料*⁷を見るとよいでしょう。

Debian パッケージのメンテナが git-buildpackage を使っている場合は gbp pq も使えます。2015 年 9 月の第 130 回東京エリア Debian 勉強会での Debian パッケージング道場の資料*⁸を見るとよいでしょう。

日本語訳を書き換えてみます。

```
$ apt-get source hello
(snip)
$ cd hello-2.10/
$ editor po/ja.po
```

変更内容を debian/changelog に書きます。

```
$ dch
$ head -n 10 debian/changelog
hello (2.10-1.1) UNRELEASED; urgency=medium

 * Non-maintainer upload.
 * po/ja.po: Refine translation.

-- YOSHINO Yoshihito <yy.y.ja.jp@gmail.com> Sat, 22 Apr 2017 12:02:56 +0900

hello (2.10-1) unstable; urgency=low

 * New upstream release.
```

debuild してみます。

*⁶ debian/source/format がでない場合は古いバージョン 1.0 です。これは Debian パッケージングの部分と上流のコードへの変更を分離して管理していないので、4.3 節と同じ作業でできます。

*⁷ <http://tokyodebian.alioth.debian.org/pdf/debianmeetingresume201302-dojo.pdf>

*⁸ <http://tokyodebian.alioth.debian.org/pdf/debianmeetingresume201509-presentation.pdf>

```

$ debuild -us -uc
dpkg-buildpackage -rfakeroot -us -uc
dpkg-buildpackage: info: source package hello
dpkg-buildpackage: info: source version 2.10-1.1
dpkg-buildpackage: info: source distribution UNRELEASED
(snip)
dpkg-source -b hello-2.10
dpkg-source: info: using source format '3.0 (quilt)'
dpkg-source: info: building hello using existing ./hello_2.10.orig.tar.gz
dpkg-source: info: local changes detected, the modified files are:
hello-2.10/po/ja.po
dpkg-source: error: aborting due to unexpected upstream changes, see /tmp/hello_2.10-1.1.diff.sbQf7t
dpkg-source: info: you can integrate the local changes with dpkg-source --commit
dpkg-buildpackage: error: dpkg-source -b hello-2.10 gave error exit status 2
debuild: fatal error at line 1116:
dpkg-buildpackage -rfakeroot -us -uc failed

```

dpkg-source --commit するとパッチ名が聞かれます。入力するとその名前のファイルに変更差分を保存されてエディタが起動します。パッチの説明を先頭に書くことになっているので、自動生成されたテンプレートに書かれている通り “DEP-3”^{*9}に従って書いてください。

```

$ dpkg-source --commit
dpkg-source: info: local changes detected, the modified files are:
hello-2.10/po/ja.po
Enter the desired patch name: refine-ja-translation
dpkg-source: info: local changes have been recorded in a new patch: hello-2.10/debian/patches/refine-ja-translation

```

debuild します。

```

$ debuild -us -uc
(snip)

```

さて、ここでさらにデフォルトの hello 出力を変えてみることにします。

```

$ editor src/hello.c

```

変更内容を debian/changelog に追記します。

```

$ dch
$ head -n 10 debian/changelog
hello (2.10-1.1) UNRELEASED; urgency=medium

* Non-maintainer upload.
* po/ja.po: Refine translation.
* src/hello.c: Change default message.

-- YOSHINO Yoshihito <yy.y.ja.jp@gmail.com> Sat, 22 Apr 2017 12:02:56 +0900

hello (2.10-1) unstable; urgency=low

```

debuild してみます。

```

$ debuild -us -uc
dpkg-buildpackage -rfakeroot -us -uc
dpkg-buildpackage: info: source package hello
dpkg-buildpackage: info: source version 2.10-1.1
(snip)
dpkg-source: info: using source format '3.0 (quilt)'
dpkg-source: info: building hello using existing ./hello_2.10.orig.tar.gz
dpkg-source: info: local changes detected, the modified files are:
hello-2.10/src/hello.c
dpkg-source: error: aborting due to unexpected upstream changes, see /tmp/hello_2.10-1.1.diff.xIhd5q
dpkg-source: info: you can integrate the local changes with dpkg-source --commit
dpkg-buildpackage: error: dpkg-source -b hello-2.10 gave error exit status 2
debuild: fatal error at line 1116:
dpkg-buildpackage -rfakeroot -us -uc failed

```

dpkg-source --commit します。何度でもできます。

```

$ dpkg-source --commit
dpkg-source: info: local changes detected, the modified files are:
hello-2.10/src/hello.c
Enter the desired patch name: change-default-message
dpkg-source: info: local changes have been recorded in a new patch: hello-2.10/debian/patches/change-default-message

```

^{*9} <http://dep.debian.net/deps/dep3/>

debuild します。

```
$ debuild -us -uc
(snip)
dh_auto_test: make -j1 check VERBOSE=1 returned exit code 2
debian/rules:3: ターゲット 'build' のレシピで失敗しました
make: *** [build] エラー 2
dpkg-buildpackage: error: debian/rules build gave error exit status 2
debuild: fatal error at line 1116:
dpkg-buildpackage -rfakeroot -us -uc failed
```

テストが通らなかったので直します。最後に保存したパッチファイルの修正が必要なので、ここからは quilt(1) の知識が必要です。詳細は 2007 年 1 月の第 24 回東京エリア Debian 勉強会資料にある “パッチ管理ツール quilt の使い方”^{*10} を見るとよいでしょう。

```
$ export QUILT_PATCHES=debian/patches
$ quilt add tests/hello-1
ファイル tests/hello-1 をパッチ debian/patches/change-default-message に追加しました
$ editor tests/hello-1
$ quilt refresh
パッチ debian/patches/change-default-message をリフレッシュしました
```

debuild して内容に満足したでしょうか。さて、ここまでバージョン番号を NMU の 2.10-1.1 にしましたが、Debian 本体でも NMU が行われて同じ 2.10-1.1 が出現してしまうかもしれません。ローカルのカスタマイズなのでもう少しバージョンを下げしておくことにします。

```
$ dch
$ head -n 10 debian/changelog
hello (2.10-1.0) UNRELEASED; urgency=medium

 * Non-maintainer upload.
 * po/ja.po: Refine translation.
 * src/hello.c: Change default message.

-- YOSHINO Yoshihito <yy.y.ja.jp@gmail.com> Sat, 22 Apr 2017 12:02:56 +0900

hello (2.10-1) unstable; urgency=low
```

debuild してみます。

```
$ debuild -us -uc
dpkg-buildpackage -rfakeroot -us -uc
dpkg-buildpackage: info: source package hello
dpkg-buildpackage: info: source version 2.10-1.0
(snip)
dpkg-genchanges > ../hello_2.10-1.0_amd64.changes
dpkg-genchanges: info: not including original source code in upload
dpkg-source --after-build hello-2.10
dpkg-buildpackage: info: binary and diff upload (original source NOT included)
Now running lintian...
W: hello source: ancient-standards-version 3.9.6 (current is 3.9.8)
Finished running lintian.
```

特に lintian 警告は出ないようなので、これでとりあえずよさそうです。dch -r して再度 debuild すれば完成です。

4.4.2 新しいバージョンなどに置き換える

uscan(1), uupdate(1) などが使えます。“新メンテナーガイド” (apt install maint-guide-ja を実行してから /usr/share/doc/maint-guide-ja/の中をみてください^{*11}) の “8. パッケージの更新” を読むとよいでしょう。

git-buildpackage を使っている場合は gbp import-orig と gbp pq も使えます。

なお、カスタマイズしているパッケージが Ruby などの言語のパッケージ (特にライブラリパッケージ) で、使っているライブラリの依存関係などが大きく更新されている場合は一からパッケージングをやり直したほうがよいこともあります。

GNU hello の最新 Git スナップショットに置き換えてみます。

^{*10} <http://tokyodebian.alioth.debian.org/pdf/debianmeetingresume200701.pdf>

^{*11} <https://www.debian.org/doc/manuals/maint-guide/index.ja.html> にもあります。


```

$ git clone https://git.savannah.gnu.org/git/hello.git
(snip)
$ cd hello/
$ ./bootstrap
./bootstrap: Bootstrapping from checked-out hello sources...
(snip)
./bootstrap: done. Now you can run './configure'.
$ ./configure
(snip)
$ make check syntax-check distcheck
(snip)
=====
hello-2.10.17-4339 archives ready for distribution:
hello-2.10.17-4339.tar.gz
=====

```

```

$ uupdate ../hello-2.10.17-4339.tar.gz
update: new version number not recognized from given filename
update: Please run uupdate with the -v option
$ uupdate ../hello-2.10.17-4339.tar.gz -v 2.10.17-4339
update: New Release will be 2.10.17-4339-1.
Symlinking to pristine source from hello_2.10.17-4339.orig.tar.gz...
update: Untarring the new sourcecode archive ../hello-2.10.17-4339.tar.gz
update: Unpacking the debian/ directory from version 2.10-1 worked fine.
update: Remember: Your current directory is the OLD sourcearchive!
update: Do a "cd ../hello-2.10.17-4339" to see the new package
$ cd ../hello-2.10.17-4339

```

debuild してみます。

```

$ debuild -us -uc
dpkg-buildpackage -rfakeroot -us -uc
dpkg-buildpackage: info: source package hello
dpkg-buildpackage: info: source version 2.10.17-4339-1
(snip)
dpkg-genchanges >../hello_2.10.17-4339-1_amd64.changes
dpkg-genchanges: info: including full source code in upload
dpkg-source --after-build hello-2.10.17-4339
dpkg-buildpackage: info: full upload (original source is included)
Now running lintian...
W: hello source: changelog-should-mention-nmu
W: hello source: source-nmu-has-incorrect-version-number 2.10.17-4339-1
W: hello source: ancient-standards-version 3.9.6 (current is 3.9.8)
Finished running lintian.

```

lintian に従って、“Debian 開発者リファレンス” も読みつつ debian/changelog を直します。あとは、まだ New upstream release でもないので直します。

```

$ dch
$ head -n 10 debian/changelog
hello (2.10.17-4339-0.1) UNRELEASED; urgency=medium

* Non-maintainer upload.
* New upstream snapshot.

-- YOSHINO Yoshihito <yy.y.ja.jp@gmail.com> Sat, 22 Apr 2017 13:01:54 +0900

hello (2.10-1) unstable; urgency=low

* New upstream release.

```

debuild します。

```

$ debuild -us -uc
dpkg-buildpackage -rfakeroot -us -uc
dpkg-buildpackage: info: source package hello
dpkg-buildpackage: info: source version 2.10.17-4339-0.1
(snip)
dpkg-buildpackage: info: full upload (original source is included)
Now running lintian...
W: hello source: ancient-standards-version 3.9.6 (current is 3.9.8)
Finished running lintian.

```

lintian は問題なさそうです。前節同様、Debian 本体でもこの Git スナップショットの NMU が出てしまうかもしれないので、さらにもう少しバージョンを下げておくことにします。

```
$ dch
$ head -n 10 debian/changelog
hello (2.10.17-4339-0.0) UNRELEASED; urgency=medium

* Non-maintainer upload.
* New upstream snapshot.

-- YOSHINO Yoshihito <yy.y.ja.jp@gmail.com> Sat, 22 Apr 2017 13:01:54 +0900

hello (2.10-1) unstable; urgency=low

* New upstream release.
```

debuild します。

```
$ debuild -us -uc
dpkg-buildpackage -rfakeroot -us -uc
dpkg-buildpackage: info: source package hello
dpkg-buildpackage: info: source version 2.10.17-4339-0.0
(snip)
dpkg-source --after-build hello-2.10.17-4339
dpkg-buildpackage: info: full upload (original source is included)
Now running lintian...
W: hello source: ancient-standards-version 3.9.6 (current is 3.9.8)
Finished running lintian.
```

lintian はとりあえず問題なさそうです。dch -r して debuild すれば完成です。

4.5 メンテナンス

カスタマイズしていたパッケージの新バージョンがそのパッケージの配布元 (Debian など) でリリースされたときは、カスタマイズ作業やり直しです。

そうではないが上流コードのさらに新しいバージョンがリリースされたときは、uscan と uupdate、gbp pq などを使えます。

4.6 まとめ

Debian ポリシーなどの慣習に従うことで、Debian システムの恩恵が受けられます。そのためにポリシーなどを本当は読む必要がありますが、debhelper を素直に使っていればある程度はポリシーなどに従って作れるので、まずは作ってみて debuild して、lintian の結果を無視しないで詳細説明も読んで従いましょう。読んでもわからないところがあったら Debian JP メーリングリストや Debian 勉強会などで質問すれば答えてくれるでしょう。

4.7 参考文献

- packaging-tutorial
- maint-guide-ja
- debian-policy
- lintian
- developers-reference-ja
- Debian Backports - Contribute <https://backports.debian.org/Contribute/>
- DEP-3: Patch Tagging Guidelines <http://dep.debian.net/deps/dep3/>
- 小林儀匡 “パッチ管理ツール quilt の使い方” 2007 年 1 月 第 24 回東京エリア Debian 勉強会資料
- 岩松信洋 “基本的なパッケージの作成方法手引書” 2013 年 2 月 Debian パッケージング道場資料
- 岩松信洋 “git build-package” 2015 年 9 月 第 130 回東京エリア Debian 勉強会 Debian パッケージング道場資料



Debian 勉強会資料

2017年4月22日 初版第1刷発行

東京エリア Debian 勉強会（編集・印刷・発行）
