Debian Pure Blends
How to work with the Blends framework

Andreas Tille

MiniDebConf Berlin 2010

Berlin, June 10, 2010
Overview

1. How to start a Blend
   - Subsetting Debian
   - Forming a team

2. Used techniques
   - Metapackages
   - Web sentinel

3. Future
Why creating a Debian Pure Blend

- Focussing users to subset of packages
- Adapting system and user interface to specific needs (working environment, user language, etc.)
- While Debian stays general support specialists as well
- **No derivative** from Debian

*Basic idea: Do not make a separate distribution but make Debian fit for special purpose instead*
Attracting people to use Blends

Users

- I18n-ed web pages displaying relevant packages
- Promoting software that builds a complete working environment
- Rise user interest by providing ready to install software in the context of their work field

Developers

- Simple way to categorise packages ("tasks files")
- Key documentation feature
- QA pages (Bugs of relevant packages)
- Acceptance of new methods higher if the techniques provided are convincing enough
A Blend is more than just technique

- Choose an interesting topic for a large user group
- Try to get people involved
  - developers
  - users
- Invite maintainers of applications fitting the scope of the Blend
- Involve upstream maintainers and active users of interesting applications (and possibly turn them into DDs)
- Example: Debian has now at least five more developers because they joined via Debian Med involvement
Special applications
Building metapackages using `blends-dev`

- Define set of dependency relations in `tasks` files
- `blends-dev` does the following automatically:
  - Verify availability of `Depends`/`Recommends`
  - Packages unavailable in `main` will be turned into `Suggests`
  - Create proper `debian/control` file to build valid metapackages
  - Create `tasksel control file` `<BLEND>-tasks.desc`
Using `blends-dev`

`debian/rules`
```bash
#!/usr/bin/make -f
include /usr/share/blends-dev/rules
```

`debian/control.stub`
- `debian/control` will be autogenerated
- Only information for source package is in stub
- See
  ```bash
  /usr/share/doc/blends-dev/examples/debian
  ```

`tasks/*`
- See `tasks` or other Blends for working examples
Tasks files

**Similar to** *debian/control*

**Task:** *taskname*

**Description:** *Shortdescription
Longdescription*

**Depends:** *some dependant packages*

**Recommends:** *some recommended packages*

**Suggests:** *some suggested packages*
Currently tasks and bugs pages
Providing information about packages of interest
Created by reading tasks files from Blends SVN containing
  • Dependency relations of packages inside Debian
  • Preliminary package information / WNPP
Gathering all available information about the package dependencies defined in the tasks file from
  • Ultimate Debian Database (UDD): (translated) descriptions, versions and architectures, screenshot URLs, bugs, etc.
  • Additional information in tasks file itself
Intention of tasks pages

- Key entry point for users
- Quick overview about what’s inside Debian regarding their specific work field
- Turned out to be QA tool for developers as well
- Meta information like
  - Homepage
  - Maintainer and VCS of Debian packaging
  - Screenshot (http://screenshots.debian.net)
  - DEHS, versions and architectures
  - DebTags
  - Popcon
  - even scientific quotation if available

→ Demo http://blends.alioth.debian.org
Additional fields in tasks files

Prospective packages

Depends: *not yet existing package name*
Homepage: *Homepage of project*
Responsible: *Future maintainer (optional)*
License: *License of software to package*
WNPP: *Bug number (optional)*
Vcs-vcstype: *Vcs URI (optional)*
Vcs-Browser: *Vcs URL (optional)*
Pkg-URL: *URL to unofficial package*
Pkg-Description: *Description of prospective package*
 Configuration of web sentinel

svn://svn.debian.org/blends/blends/trunk/webtools/webconf/

Blend: debian-med
ProjectName: Debian Med
ProjectUrl: http://debian-med.alioth.debian.org/
Homepage: http://www.debian.org/devel/deb...
AliothUrl: http://alioth.debian.org/project...
ProjectList: debian-med@lists.debian.org
LogoUrl: http://debian-med.alioth.debian.org/
OutputDir: /var/lib/gforge/chroot/home/gro...
DataDir: /var/lib/gforge/chroot/home/gro...
VcsDir: /svn/blends/projects/med/trunk/...
CSS: ../inc/style.css
Advertising: _('Help us to see Debian used by...
PkgList: debian-med-packaging@lists.alio...
Weighting bugs

- Try to find a measure for bugs of dependant packages
- Currently not normalised to the number of dependencies but rather regarding absolute number of bugs
- Weighting numbers for the different severities ranging from 10 for the RC bugs until 0 for wishlist bugs

Example calculation

1 serious bug in dependent pkg: $1 \times 10 \times 3 = 30$
2 important bugs in dependent pkg: $2 \times 5 \times 3 = 30$
1 important bug in suggested pkg: $1 \times 5 \times 1 = 5$
1 normal bug in dependent pkg: $1 \times 3 \times 3 = 9$
1 minor bug in dependent pkg: $1 \times 1 \times 3 = 3$

weighted sum = 77
Metapackage can not be in status "good" if there is at least serious (or higher) bug in a dependant package

- Not "very good" if there is a RC bug in a suggested package
- Two RC bugs in suggested packages might qualify for "good" - if there are only a very view other bugs
Planned features for web sentinel

- More QA overviews
  - Lintian report overview
  - Adding Ubuntu bugs
- i18n information of applications
- Upstream metadata like
  - Please cite
  - Link to donation page
Make `blends-dev` use UDD

- Build metapackages based on UDD information
- Thus enabling `architecture=any` metapackages
- Include tasks file information into UDD
Try to establish technique

- Further enhancements
- Rewrite `blends-dev` to use UDD
- Make even more projects like DebiChem and Debian-GIS actively using the framework
- Try to bring back external projects to Debian by providing attractive tools
This talk can be found at
http://people.debian.org/~tille/talks/
Andreas Tille <tille@debian.org>