Workshop *cdd-dev*

Building CDD meta packages easily

Andreas Tille

Extremadura Workshop

Merida, Oct 18, 2007

Structure

1. Overview CDD features
   - Meta-packages
   - *`tasksel`*
   - User menus
2. Using *`cdd-dev`*
   - Task files
   - *`CDD-config`*
   - User menus
3. Further ideas
   - Web tools
   - Live CD
   - FAI
   - DebTags
   - Increasing menus

What are packages

- Main part of the work of a distributor
- Contains
  - Executable programs
  - configuration
  - `{pre/post}install` + `{pre/post}remove` scripts
  - Relations to other packages

  ⟶ More than just an archive of files
- Enables easy handling
  - Ensures compliance of all dependencies
  - Clean upgrades
  - Easy handling of security updates

  ⟶ Basis of modern GNU/Linux distributions

Problem: Increasing number of packages

Meta-Packages

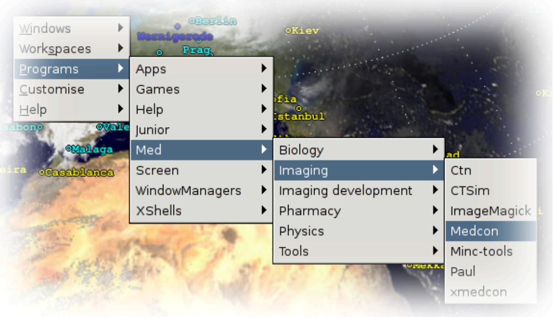
- *Special packages* with following content:
  1. Relations on other Debian packages (essential)
     - "Depends" for necessary packages
     - "Recommends" for interesting packages
     - "Suggests" for not so important or non-free packages
  2. Menu entries (recommended)
     - Customisation of default entries possible
     - Documentation entry if useful
  3. Adapted configuration (optional)
     - *debconf* questions or pre-seeding
     - *cfengine* scripts
- Special meta package: `<cdd>`*-config*
- Meta-packages ⟶ kind of task packages
- Easily built by using *cdd-dev* package

tasksel = first user interface

- First chance to select sets of packages
- CDD is about selecting sets of packages
- *cdd-dev* installs
  /usr/share/tasksel/debian-CDD-tasks.desc

Role based user menus

- User with role *<cdd>* gets extra user menu



- User with role *<cdd>* is provided with additional menu
- Do not bother other users with those additional menus
- Implemented in current Debian-Med packages
- Role system is implemented by flexible plugins

Short introduction to build meta-packages

Control file

```
$ cp -a /usr/share/doc/cdd-dev/examples/tasks .
$ cat tasks/README
$ edit tasks/task1
Task: task name
Description: short description
 long description as in any debian/control file
Depends: dependency1, dependency2, ...
Suggests: suggests1, suggests2, ...  (optional)
$ cp tasks/task1 tasks/<meta-package-name>
```

For each meta package this skeleton of a `debian/control` *entry is needed*

Working example: *`apt-get source debian-med`*

Short introduction to build meta-packages (2)

Directory debian

```
$ cp -a /usr/share/doc/cdd-dev/examples/debian .
$ cat debian/README
$ edit debian/control.stub
 change the variables of the example
```

debian/rules includes common *rules* file

debian/TASK.{install,manpages,...} if needed

cdd-gen-control

- Examine *sources.list* file matching distribution that is mentioned in *debian/changelog* if not explicitely given via `-s` option
- If no *sources.list* file is provided by package source files in */etc/cdd/* are used as default
- Each task file is added to *debian/control* where the `Depends:` and `Recommends:` from the task file are added as `Recommends:`
- `Suggests:` and packages not available according to *sources.list* are added as *Suggests:*
- Can be called via *make* (use the Makefile */usr/share/doc/cdd-dev/examples/Makefile*)

Meta package *`CDD-config`*

Optional

```
$ cp -a /usr/share/doc/cdd-dev/examples/config .
$ cat config/README
$ edit config/{conf,control,config.1}
 change variable _CDD_
```

Initialisation of role based menu system

- All meta packages will depend on the config package
- Installs *`auto-apt`* helper
- Strongly suggested

Providing user menus

Optional

```
$ cp -a /usr/share/doc/cdd-dev/examples/menu .
$ cat menu/README
$ cat menu/task1/README
$ edit menu/task1/dep1.menu
$ edit menu/task1/dep2.txt
$ edit menu/task1/dep3.html
```

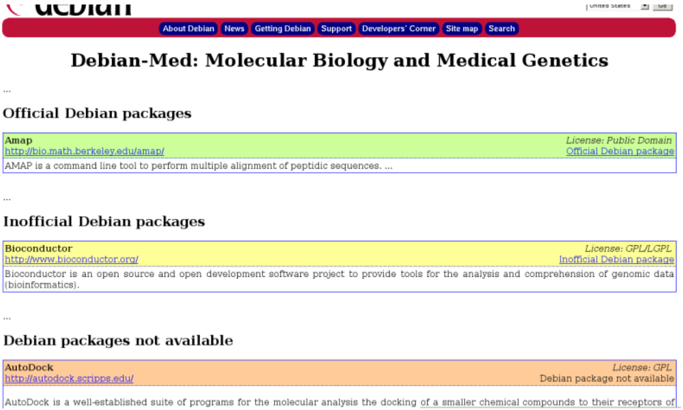
Create valid menu entries with sensible descriptions

Find task names

```
$ cp menu/task1 menu/<meta package name>
```

User menu entries for every dependencies - at least pointing to documentation

Generate package overview pages



Debian-Med: Molecular Biology and Medical Genetics

...

Official Debian packages

Amap *License: Public Domain*
http://bio.math.berkeley.edu/amap/ Official Debian package

AMAP is a command line tool to perform multiple alignment of peptidic sequences. ...

...

Inofficial Debian packages

Bioconductor *License: GPL/LGPL*
http://www.bioconductor.org/ Inofficial Debian package

Bioconductor is an open source and open development software project to provide tools for the analysis and comprehension of genomic data (bioinformatics).

...

Debian packages not available

AutoDock *License: GPL*
http://autodock.scripps.edu/ Debian package not available

AutoDock is a well-established suite of programs for the molecular analysis the docking of a smaller chemical compounds to their receptors of

http://www.debian.org/devel/debian-med/microbio

Extending task file syntax

Adding not yet integrated software

`Depends:` dialign-t

`Homepage:` *http://dialign-t.gobics.de/*

`License:` LGPL

`WNPP:` *445983* (optional)

`Pkg-URL:` URL to inofficial package (optional)

`Developer:` Who is responsible (optional) ??

`Pkg-Description:` *multiple sequence alignment*
DIALIGN-T is a command line tool to perform
multiple alignment of protein or DNA
sequences. It is a complete reimplementation
...

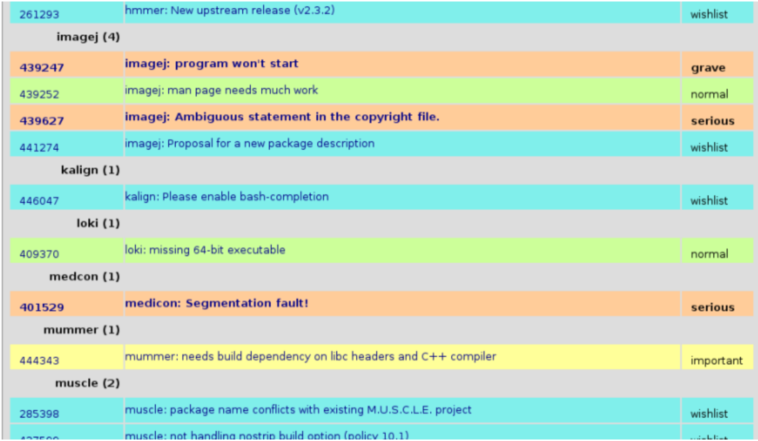
Advantage of listing non existent packages

- `Depends`/`Recommends` are turned into `Suggests` for non existent packages so there is no harm in listing non existent packages
- Once the package exists the meta package works automatically - at least for users who regard `Suggests`
- No update of meta packages required once new packages become available
- Next version of meta packages will regard the new package automatically - maintainer will not forget new packages
- Users of inofficial packages also profit
- No drawbacks for packages that do not exist at all

Problems and alternative

- `Pkg-Description` versus `Description`
- Current syntax allows arbitrary number of Depends per line
- Weak syntax
- Alternative “WNPP”-task

Bug overview pages



261293	hmmer: New upstream release (v2.3.2)	wishlist
imagej (4)		
439247	**imagej: program won't start**	**grave**
439252	imagej: man page needs much work	normal
439627	**imagej: Ambiguous statement in the copyright file.**	**serious**
441274	imagej: Proposal for a new package description	wishlist
kalign (1)		
446047	kalign: Please enable bash-completion	wishlist
loki (1)		
409370	loki: missing 64-bit executable	normal
medcon (1)		
401529	**medicon: Segmentation fault!**	**serious**
mummer (1)		
444343	mummer: needs build dependency on libc headers and C++ compiler	important
muscle (2)		
285398	muscle: package name conflicts with existing M.U.S.C.L.E. project	wishlist
[illegible]	muscle: not handling nostrip build option (policy 10.1)	wishlist

http://debian-med.alioth.debian.org/bugs.php

Code: `/var/lib/gforge/chroot/home/groups/debian-med/scripts/update-bugs`

Building live CDs using `debian-live`

- `debian-live` makes building live CDs easy
- Lacking documentation and finalized API
- Using a common wrapper for CDDs might simplify process
- Perhaps just a quick `debian-live` intro in `cdd-doc`

Using FAI in CDDs

- FAI sounds interesting for several CDDs
- Chances for some common profile are low
- Perhaps just a quick `fai` intro in `cdd-doc`

Using DebTags in CDDs

- Every CDD should have a close look onto their packages that they are nicely debtagged
- DebTags based installation?
- Drawback: Possible vandalism in DebTags database

Freedesktop.Org and general menu

- Integration of Freedesktop.Org
- Enable general CDD menu instead of user menu

This talk can be found at
http://people.debian.org/˜ tille/talks/

Andreas Tille <tille@debian.org>