



# Debian Packaging

Sebastian Reichel <[sre@debian.org](mailto:sre@debian.org)>

- Softwaresuche
- Abhangigkeiten
- "Automatisches" updaten
- ...

- 
- apt-get / aptitude
    - ▶ apt-cache search foo
    - ▶ apt-get install foo
    - ▶ apt-get remove foo

- dpkg
  - ▶ dpkg -i foo.deb
  - ▶ dpkg -r foo
  - ▶ dpkg -l
  - ▶ dpkg -L foo

# Software mittels Quellcode installieren

- 1 Finde Software, die noch nicht paketiert ist
- 2 Downloade das Quellcode-Archiv vom letzten Release oder Snapshot
- 3 Entpacke das Archiv
- 4 Führe “./configure” aus
- 5 Führe “make” aus
- 6 Führe “sudo make install” aus
- 7 Konfiguration

# Debian Packaging Workflow

- 1 Quellcode-Archiv herunterladen und entpacken
- 2 Das Quellcode-Archiv sollte "paketname\_paketversion.orig.tar.gz" heißen
- 3 In dem entpackten Code ein Verzeichnis "*debian*" anlegen
- 4 In diesem Verzeichnis werden im folgenden diese Dateien angelegt:
  - ▶ *debian/changelog* - Changelog
  - ▶ *debian/control* - Metainformationen (Abhängigkeiten, Beschreibungen, ...)
  - ▶ *debian/copyright* - Copyright & Lizenz Informationen
  - ▶ *debian/watch* - URL um automatisch auf neue Versionen zu prüfen
  - ▶ *debian/rules* - Informationen wie der Quellcode gebaut werden soll

# debian/changelog

```
0xfffff (0.6~git20130421-1) UNRELEASED; urgency=low
```

- \* New upstream snapshot
- \* manpage is now generated upstream

```
-- Sebastian Reichel <sre@debian.org> Sat, 01 Jun 2013 23:42:00 +0200
```

```
0xfffff (0.6~git20130406-1) unstable; urgency=low
```

- \* Initial release. (Closes: #514244)

```
-- Sebastian Reichel <sre@debian.org> Sat, 06 Apr 2013 17:21:36 +0200
```

```
Source: 0xfffff
Section: misc
Priority: extra
Maintainer: Sebastian Reichel <sre@debian.org>
Standards-Version: 3.9.4
Build-Depends: debhelper (>= 9),
                 libusb-dev
Homepage: http://gitorious.org/0xfffff/
Vcs-Git: git://git.debian.org/git/pkg-n900/0xfffff.git
Vcs-Browser: http://git.debian.org/?p(pkg-n900/0xfffff.git;a=summary
```

```
Package: 0xfffff
Architecture: any
Multi-Arch: foreign
Depends: ${misc:Depends}, ${shlibs:Depends}
Description: Open Free Fiasco Firmware Flasher
The 'Open Free Fiasco Firmware Flasher' aka 0xFFFF utility implements
a free userspace handler for the NOLO bootloader and extended features
for flashing, dumping and getting information for the Nokia Internet
Tablets. Thus it's a free alternative to Nokia's closed source flashing
utility called flasher.

.
Currently supported are the Nokia 770, N800, N810, N810w and the N900.
```

Format: <http://www.debian.org/doc/packaging-manuals/copyright-format/1.0/>

Source: <http://gitorious.org/0xffff/>

Files: \*

Copyright: 2007-2012 pancake <pancake@youterm.com>

2011 Michael Buesch <mb@bu3sch.de>

2011-2013 Pali Rohar <pali.rohar@gmail.com>

License: GPL-3+

LICENSE TEXT

Files: debian/\*

Copyright: 2013 Sebastian Reichel <sre@debian.org>

License: ISC

LICENSE TEXT

# Debian Free Software Guidelines (DFSG)

- [http://www.debian.org/social\\_contract#guidelines](http://www.debian.org/social_contract#guidelines)
- Erlaubnis der unbeschränkten Weitergabe
- Verfügbarkeit des Quellcodes
- Erlaubnis, die Software zu ändern und abgeleitete Werke zu erstellen.
- Keine Diskriminierung von Personen oder Personengruppen
- Keine Beschränkung des Einsatzbereichs (zum Beispiel kommerzielle Nutzung)
- Keine Einschränkung bezogen auf andere Programme
- Die Rechte der Lizenz gelten für alle (insbesondere unabhängig davon, ob die Software als Teil der Debian-Distribution weitergegeben wird oder nicht)

```
version=3

opts=\
downloadurlmangle=s/\//trees\/\//archive-tarball\//g,\ \
filenamemangle=s/^.*\//(\d+\[\d\.\]+\)$/0xffff-$1.tar.gz/, \
uversionmangle=s/-beta(\d+)/~beta$1/ \
http://gitorious.org/0xffff/0xffff/trees/master \
/0xffff/0xffff/trees/(\d+\[\d\.\]+\-\beta\*\d*)
```

```
version=3
http://sf.net/nesc/nesc-(.+)\.tar\.gz
```

# debian/rules

```
#!/usr/bin/make -f

%:
 dh $@

override_dh_auto_install:
 dh_auto_install -- PREFIX=/usr
```

- <http://manpages.debian.net/cgi-bin/man.cgi?query=debhelper>

## debian/{preinst,postinst,prerm,postrm}

```
#!/bin/sh
set -e

if [ "$1" = "configure" ]; then
    # do post installation configuration here
    # (use debconf for user interaction)
fi

#DEBHELPER#

exit 0
```

# dpkg-buildpackage

- dpkg-buildpackage
- dpkg-buildpackage -b
- dpkg-buildpackage -S

- `pbuilder --create`
- `pbuilder --update`
- `pbuilder --build file.dsc`

# Ergebnis überprüfen

- `lintian --display-info --pedantic file.changes`
- `dpkg -c file.deb`
- `dpkg --info file.deb`

# debsign & dput

- debsign file.changes
- dput file.changes

# Wer darf Pakete in Debian maintainen?

- Technisch darf jeder **Debian Developer** jedes beliebige Paket bearbeiten
- Es gibt auch so genannte **New Maintainer**, welche nur für bestimmte Pakete freigeschaltet sind
- Pakete können auch **gesponsort** werden (d.h. ein DD signiert das Paket von jemand anderem und lädt es hoch)
- <http://mentors.debian.net/>

# Paket Maintainergruppen

- <http://git.debian.org>
- git-import-orig
- git-buildpackage
- pristine-tar

- Debian Policy  
<http://www.debian.org/doc/debian-policy/>
- Debian New Maintainer's Guide  
<http://www.debian.org/doc/manuals/maint-guide/>
- Debian Developer's Reference  
<http://www.debian.org/doc/manuals/developers-reference/>



Fragen?