

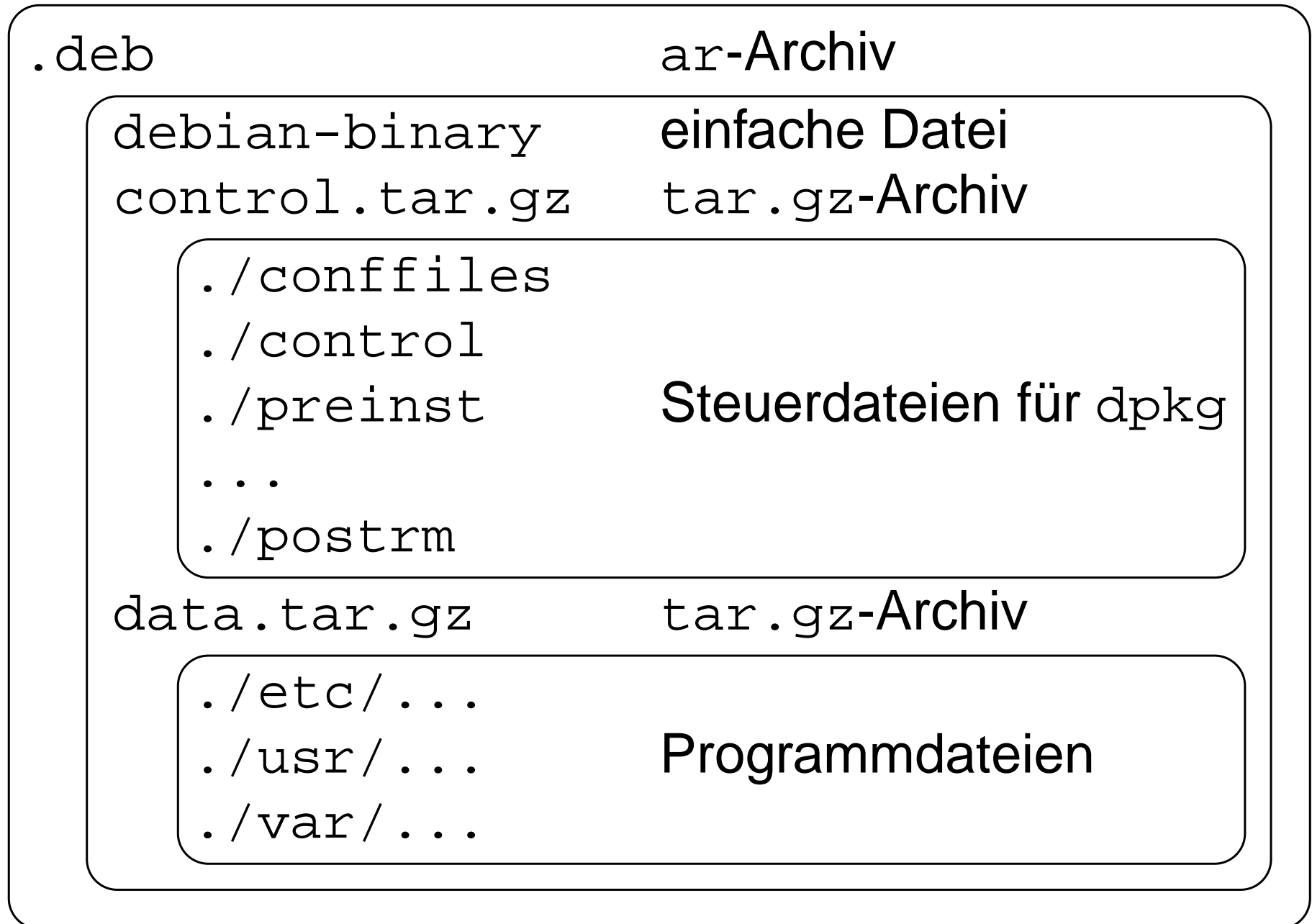
.deb – Das Debian-Paket-Format

Jörg Sommer <joerg@alea.gnuu.de>

16. Juni 2005

Ein .deb von innen

Container-
prinzip:



Ein `.deb` von innen (2)

- verwendbar mit Standard-Unix-Tools
- Dateien mit relativer Pfadangabe in `.tar.gz`-Archiven machen Installation in anderen Pfaden möglich: `dpkg-deb -x $PKG.deb /tmp`

Informationen über ein `.deb`

`dpkg-deb` (wird auch durchgereicht von `dpkg`)

-c: Inhalt von `data.tar.gz` auflisten

-f: `control` (aus `control.tar.gz`) anzeigen

-I: wie `-f` plus einige Statistiken über `control.tar.gz`

-e: `control.tar.gz` in ein Unterverzeichnis `DEBIAN` auspacken

-x / -X: `data.tar.gz` in ein angegebenes Verzeichnis auspacken

Klassen von Quellpaketen

Quellpaket = Verbund aus `.orig.tar.gz`, `.dsc` (und `.diff.gz`)

single es wird nur ein `.deb` erzeugt

multi es werden mehrere `.debs` erzeugt

library es wird ein Paket mit einer Bibliothek und ein dazugehöriges Entwicklungspaket (`*-dev*` `.deb` erzeugt

kmod es wird ein Paket für Kernelmodule und ggf. ein Paket mit Dokumentation oder Zusatzprogrammen erzeugt

native direkt als `.deb` entwickelt (z. B. `apt-get`, `dpkg`, `aptitude`); hat kein `.diff.gz`

Vom `.tar.gz` zum `.deb`

`dh_make` aus dem Paket `dh-make`

- Aufruf im ausgepackten Verzeichnis
- **Achtung:** Verzeichnisname in der Form `<package>-<version>`
- Option `-f` mit dem Upstream-`.tar.gz`
- `-c` Lizenz von `artistic`, `bsd`, `gpl`, `lgpl`; andere Lizenz von Hand setzen

Das Beispiel: Bootchart

- <http://www.bootchart.org/>
- gestartet nach einem [Posting von Owen Taylor](#) auf der Fedora-Mailing-Liste

The challenge is to create a single poster showing graphically what is going on during the boot, what is the utilization of resources, how the current boot differs from the ideal world of 100% disk and CPU utilization, and thus, where are the opportunities for optimization.

- Skript zum Aufzeichnen der Systemaktivität während des Bootvorgangs (`init`-Start bis Login-Prompt)
- Java-Programm zur Darstellung der Daten als SVG, PNG oder EPS

dh_make am Beispiel Bootchart

```
bootchart-0.8$ dh_make --multi --file ../bootchart-0.8.tar.gz \  
--copyright gpl
```

```
Maintainer name : Jörg Sommer  
Email-Address   : joerg@alea.gnuu.de  
Date            : Wed, 15 Jun 2005 16:12:51 +0200  
Package Name    : bootchart  
Version         : 0.8  
License         : gpl  
Type of Package : Multi-Binary  
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 bootchart Makefiles install into \$DESTDIR and not in / .

Was ist passiert?

- ../bootchart_0.8.orig.tar.gz angelegt
- debian/ mit Hauptdateien und Beispielen erzeugt

bootchart-default.ex	cron.d.ex	manpage.xml.ex
bootchart.doc-base.EX	dirs	menu.ex
bootchart-doc.docs	docs	postinst.ex
bootchart-doc.install	emacsen-install.ex	postrm.ex
changelog	emacsen-remove.ex	preinst.ex
compat	emacsen-startup.ex	prerm.ex
conffiles.ex	init.d.ex	README.Debian
control	manpage.1.ex	rules
copyright	manpage.sgml.ex	watch.ex

Hauptdateien müssen in einem Quellpaket vorhanden sein.

debian/copyright

- beinhaltet die Lizenz des Paketes; später in `/usr/share/doc/<package>/`
- Kopf

```
This package was debianized by Jörg Sommer <joerg@alea.gnuu.de>  
on Wed, 15 Jun 2005 16:43:20 +0200.
```

```
It was downloaded from <fill in ftp site>
```

```
Copyright Holder: <put author(s) name and email here>
```

Werte sind entsprechend auszufüllen:

```
It was downloaded from http://www.bootchart.org/download.html
```

```
Copyright Holder: Ziga Mahkovec <ziga.mahkovec@klika.si>
```

debian/control

```
Source: bootchart
Section: unknown
Priority: optional
Maintainer: Jörg Sommer <joerg@alea.gnuu.de>
Build-Depends: debhelper (>= 4.0.0)
Standards-Version: 3.6.1

Package: bootchart
Architecture: any
Depends: ${shlibs:Depends}, ${misc:Depends}
Description: <insert up to 60 chars description>
  <insert long description, indented with spaces>

Package: bootchart-doc
Architecture: all
Description: Documentation for bootchart
  <insert long description, indented with spaces>
```

Felder sind in der Debian-Policy Abschnitt 5.6 festgelegt

die geänderte debian/control

```
Source: bootchart
Section: admin
Priority: optional
Maintainer: Jörg Sommer <joerg@alea.gnuu.de>
Build-Depends: debhelper (>= 4.0.0), ant, java-compiler | kaffe | sablevm
Standards-Version: 3.6.1
```

```
Package: bootchart
Architecture: all
Depends: bash
Recommends: bootchart-view
Description: Boot process performance analyser
  Bootchart is a tool for performance analysis and visualization of the
  GNU/Linux boot process. Resource utilization and process information are
  collected during the boot process and are later rendered in a PNG, SVG or
  EPS encoded chart.
  .
  You can render the data with bootchart-view or at
  http://www.bootchart.org/download.html
```

```
Package: bootchart-view
Architecture: all
Depends: java-virtual-machine | kaffe
Recommends: bootchart
Suggests: gqview
Description: Boot process performance analyser (visualisation)
  -- wie oben --
  .
  This package ships the engine to render to data as graphic.
```

debian/changelog

- bearbeitbar mit dch (debchange)

- ```
bootchart (0.8-1) unstable; urgency=low

* Initial release Closes: #nnnn (nnnn is the bug number of
 your ITP)

-- Jörg Sommer <joerg@alea.gnuu.de> Wed, 15 Jun 2005 16:43:20
```

**0.8-1** Versionsnr.: Upstream–Debian; bei native Paketen nur Upstream  
**unstable** Distribution: unstable, experimental, stable (§5.6.13)

**urgency=...** Dringlichkeit dieser Version (Empfehlung für Benutzer):  
low, medium, high; Fix von Sicherheitsbugs z. B. high (§5.6.16)

- Änderungen werden Punkt (\*) für Punkt eingetragen
- Closes: #nnn sind spezielle Anweisungen für das BTS
- Versionen werden gestapelt; aktuelle am Anfang

# debian/rules

- ist für den Bau des Paketinhalts verantwortlich
- Debian-Policy Abschnitt 4.8
- muss ein `make`-Skript sein; muss ausführbar (`x`-Recht) sein
- Erforderlichen `make`-Ziele: `clean`, `binary`, `binary-arch`, `binary-indep`, `build`

```
bootchart-0.8/debian/rules
```

```
configure:
 # There is nothing to configure

build: build-stamp
build-stamp:
 dh_testdir
 ant -noinput $(ANT_OPT) jar
 touch build-stamp

clean:
 dh_testdir
 rm -f build-stamp
 -ant -noinput clean
 dh_clean
```

```
install: build-stamp
```

```
dh_testdir
```

```
dh_testroot
```

```
dh_clean -k -i
```

```
dh_installdirs -i
```

```
dh_install -i
```

```
binary binary-indep: build-stamp install
```

```
dh_testdir
```

```
dh_testroot
```

```
dh_installchangelogs ChangeLog
```

```
dh_installdocs
```

```
dh_installexamples
```

```
dh_link
```

```
dh_compress
```

```
dh_fixperms
```

```
dh_installdeb
```

```
dh_gencontrol
```

```
dh_md5sums
```

```
dh_builddeb
```

```
binary-arch:
```

```
.PHONY: build clean binary-indep binary-arch binary install \
install-indep configure
```

# Debhelper – die Heinzelmännchen bei Debian

- für viele kleine immer wiederkehrende Aufgaben existieren `dh_*`-Programme
- Beispiele: `dh_installinfo`, `dh_installetec`, `dh_installdata`, `dh_installdata`, `dh_installdata`, `dh_installdata`, `dh_installdata`, `dh_installdata`, `dh_installdata`
- Vorteil:
  - weniger (Denk-)Arbeit; wohin gehören Beispieldateien?
  - Policy-Konformität wird sichergestellt
  - zusätzliche Dinge werden erledigt; z. B. Registrierung der `.info`-Datei nach der Installation
  - mehr Sicherheit; `dh_fixperms` sorgt für korrekte Rechte
- Konfiguration per Kommandozeile oder Datei in `debian/`
- Manpages: `dh_*` und `debhelper`

# Debhelper für Bootchart

bootchart.dirs

```
mnt/bootchartd/
var/log/
```

bootchart-view.examples

```
examples/*
```

bootchart.docs

```
README
README.logger
TODO
```

bootchart-view.docs

```
README
```

bootchart.install

```
script/bootchartd sbin/
script/bootchartd.conf etc/
```

bootchart-view.install

```
bootchart.jar usr/\
 share/bootchart-view/
debian/bin/bootchart usr/bin/
```

bootchart.postrm

```
case "$1" in
 purge) rm -f /var/log/bootchart.tgz ;;
 remove|upgrade|failed-upgrade|abort-install|abort-upgrade|disappear) ;;
 *) echo "postrm called with unknown argument \"$1\"" >&2
 exit 1 ;;
esac

#DEBHELPER#
```

# Besonderheiten bei Bootchart

- `script/bootchart` ist für Debian nicht einsetzbar, da keine `java-functions` existiert und nicht verwendbar mit `sablevm`; eigene geschrieben `debian/bootchart`
- `install.sh` ist nicht brauchbar, da sich `$ROOT` nicht ohne Ändern der Datei setzen lässt; alles selbst in `rules` gemacht
- `bootchart.postrm` löscht `/var/log/bootchart.tgz`, falls Paket „ge-purged“ wird

# debian/changelog von Bootchart

```
bootchart (0.8-1) unstable; urgency=low
```

```
* Initial release
```

```
* Created an own /usr/bin/bootchart, because the upstream
version is not usable with Debian. Debian misses
java-functions and sablevm need to be called different.
```

```
-- Jörg Sommer <joerg@alea.gnuu.de> Wed, 15 Jun 2005 16:43:20
```

# Der Bau der Pakete

- In der Regel sollte das Paket `build-essential` (`gcc`, `g++`, `make`, `libc6-dev`, `dpkg-dev`) installiert sein
- gebaut wird mit `dpkg-buildpackage` (aus `dpkg-dev`)
- man braucht Root-Rechte, damit die Dateien im `.deb` die richtigen Eigentümer haben → Realisierung mit `fakeroot`

## Binärpakete:

```
bootchart-0.8$ dpkg-buildpackage -uc -b -rfakeroot
```

## Quellpakete:

```
bootchart-0.8$ dpkg-buildpackage -us -s -rfakeroot
```

## Aufräumen:

```
bootchart-0.8$ fakeroot debian/rules clean
```

# Zusammenfassung

1. `.tar.gz` auspacken und in das Verzeichnis wechseln
2. `dh_make` aufrufen
3. `debian/copyright` bearbeiten
4. `debian/control` bearbeiten
5. `debian/rules` bearbeiten
6. `debian/changelog` bearbeiten
7. `dpkg-buildpackage -rfakeroot`

# ein weiterer Vortrag

- `(post|pre)(rm|inst), configure`
- `debhelper` ausführlicher
- Patches am Paket (`dpatch`)
- Debians `update-*`-Programme
- Policy-Konformität mit `lintian` oder `linda`

# Informationsquellen

- Debian-Policy, Paket `debian-policy`
- Developers-Reference, Paket `developers-reference(-de)`
- New maintainer guide, Paket `maint-guide(-de)`
- das Paket `hello`
- Mailing-Liste [debian-devel@lists.debian.org](mailto:debian-devel@lists.debian.org)
- [Repository-Howto](#)

We are the apt.  
Resistance is futile.  
You will be packaged.  
*[dPKG]*

Danke schön!

# History

## 19. Juni 2005

- den Aufruf von `debian/rules` mit `fakeroot`; die Skripte sind zwar so gestaltet, dass es ohne geht, aber in der Regel muss `fakeroot` sein
- die Pfade für das `VerbatimInput` geändert, da die Dateien jetzt lokal in den Quellen liegen.
- die Zusammenfassung hinzugefügt
- vor jeden `dpkg-buildpackage`-Aufruf geschrieben, was er macht