

*

Gentoo Workshop Troisdorf

Sebastian Pipping <*sping@gentoo.org*>

2015-08-01

Version 1.0 | Licensed under CC-BY-SA 4.0

THEMEN

- Was leisten Ebuilds?
- Ablauf Paketierung
- ▶ Demo
 - Overlay einrichten
 - Echtes Ebuild entwickeln
- Werbung

WAS LEISTEN EBUILDS?

(Im Vergleich zu "make install")

- Reproduzierbarkeit, "Dokumentation"
- Deinstallation gratis
- Unterstützung durch Eclasses
- Kollaboration mit anderen
- Weniger Wildwuchs in `/usr/local/`

ABLAUF PAKETIERUNG

- ▶ <http://gpo.zugaina.org/> checken
 - Download URL(s) besorgen
 - Ebuild entwickeln
 - Build-System verstehen
 - Abhängigkeiten sammeln
 - Lizenzen ausfindig machen
 - Patches anfertigen
 - Ebuild/Overlay veröffentlichen

IM TERMINAL

```
mkdir media-sound/cava
cd media-sound/cava
cp TEMPLATE cava-0.3.2.ebuild
while ! ready; do
    ${EDITOR} cava-0.3.2.ebuild
    ebuild cava-0.3.2.ebuild \
        manifest clean install
done
repoman full
```

EBUILD COMMAND

```
# ebuild cava-0.3.2.ebuild \  
    manifest [clean] UP_TO_ACTION
```

ACTIONS / EBUILD FUNCTIONS (Auszug)

```
# ebuild cava-0.3.2.ebuild \  
    manifest [clean] UP_TO_ACTION
```

ACTION	FUNCTION
1. setup	→ pkg_setup
2. fetch	
. unpack	→ src_unpack
. prepare	→ src_prepare
. configure	→ src_configure
. compile	→ src_compile
. test	→ src_test
8. install	→ src_install

MEINE SHELL-FUNKTIONEN

In ~/.bashrc:

```
emci() {(
    set -x
    ebuild "$1" manifest clean install
)}
```

```
ecmc() {(
    set -x
    sudo ebuild "$1" manifest clean merge clean
)}
```

In der Shell:

```
# emci cava-0.3.2.ebuild
```


DEMO

- ▶ Git-Overlay anlegen
 - Unpriviligierten User freischalten
 - Ebuild-Template vorstellen
 - cava-Ebuild zum Laufen bekommen

ÜBERSICHT ECLASSES (aktuell 215 Eclasses)

- Programmiersprachen — python*, ruby*, ...
- Build-Systeme — cmake-utils, distutils-r1, ...
- (Live- und) Snapshot-Ebuilds — bzip, git-r3, ...
- Desktop environments — gnome*, xfconf, ...
- ...

SPEZIELLE ERWÄHNUNG VERDIENEN...

- `autotools` — `eautoreconf`
- `eutils` — `epatch`, `make_desktop_entry`
- `flag-o-matic` — Compile flags filtern
- `games` — für Spiele
- `linux-info` — Kernel config abfragen
- `multilib` — `get_libdir(lib32/lib64)`
- `toolchain-funcs` — `tc-getCC`
- `unpacker` — `unpack_deb`
- `versionator` — `get_version_component_range`

EBUILD HELPERS (außerhalb von Eclasses)

/usr/lib64/portage/python2.7/ebuild-helpers/*

die dobin doconfd dodir dodoc doenvd doexe dohard
doheader dohtml doinfo doinitd doins dolib dolib.a
dolib.so doman domo dosbin dosed dosym ecompress
ecompressdir eerror einfo elog emake eqawarn ewarn
fowners fperms keepdir newbin newconfd newdoc newenvd
newexe newheader newinitd newins newlib.a newlib.so
newman newsbin portageq prepall prepalldocs
prepallinfo prepallman prepallstrip prepinfo prepman
prepstrip

/usr/lib64/portage/python2.7/phase-helpers.sh

best_version diropts docinto docompress econf
einstall exeinto exeopts has_version insinto insopts
into libopts unpack use use_enable useq usev use_with

DEMO

- cava-Ebuild *ernst nehmen*

HÄUFIGE UPSTREAM-FEHLER

- Sandbox violations (\$DESTDIR)
- Falsche Annahmen über Pfade
- Downloads während Kompilierung/Tests
- Gebündelte Abhängigkeiten
- Feste CFLAGS/LDFLAGS
- Parallel make issues (make -j5)
- Ungefragtes Stripping von Binaries

DOWNSTREAM *DON'Ts*

- ▶ Niemals beim User Schaden anrichten
Fail-Beispiel: `rm -R "${typo}"/lib/`
- Nicht als root Ebuilds entwickeln
- Keine Tarballs, Bilder oder Binärdateien in Overlays ablegen

DEBUGGING

Für eigenen Code:
Bash-Bordmittel.

Zum Debuggen von Eclasses:

```
# ECLASS_DEBUG_OUTPUT=on ebuild ...
```

Außerdem geschrieben wird:

```
/var/tmp/portage/${CATEGORY}/${PF}/  
temp/eclass-debug.log
```


CHECKLISTE (für diesen Vortrag)

- Lddtree für Abhängigkeiten demonstriert?
- "|| die" und "nonfatal .." erwähnt?
- Eine Eclass im Texteditor geöffnet?
- Patch(es) für die Makefile eingebunden?
- Von Live-Ebuilds abgeraten?

WEITERFÜHRENDES MATERIAL

Gentoo Development Guide

<https://devmanual.gentoo.org/>

Project Sunrise

<https://wiki.gentoo.org/wiki/Project:Sunrise>

#gentoo-dev-help on Freenode IRC

WERBUNG

- ▶ Regelmäßige Updates
 - `app-admin/lib_users`
 - tmpfs auf `/var/tmp/portage`
 - `app-portage/porticron`
 - `app-portage/overlint`
 - `dev-util/image-bootstrap::betagarden`
 - `dev-python/pychroot`
 - `dev-util/meld`
 - `x11-misc/grub2-theme-preview`
 - `app-office/pinpoint`

FRAGEN!