



# patch management with DVCS

Thomas Koch

YMC AG

June 10, 2010

# Outline

Packaging mit GIT

Patches in Branches



# Debian Packages

bestehen aus

- original upstream tarball
- debian/ Verzeichnis mit scripten, controlfiles
- patches

# GIT layout

## 3 GIT-Branches

- ◉ upstream - enthält unveränderten upstream
- ◉ master - upstream + debian/
- ◉ pristine-tar - Infos zum widerherstellen des original tarballs

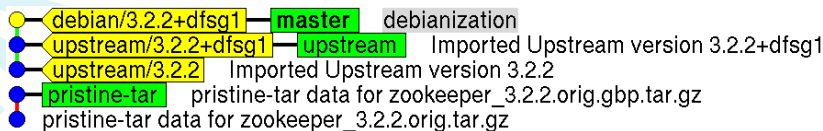
# git-import-orig

```
git-import-orig --pristine-tar --no-merge \  
../zookeeper_3.2.2.orig.tar.gz
```

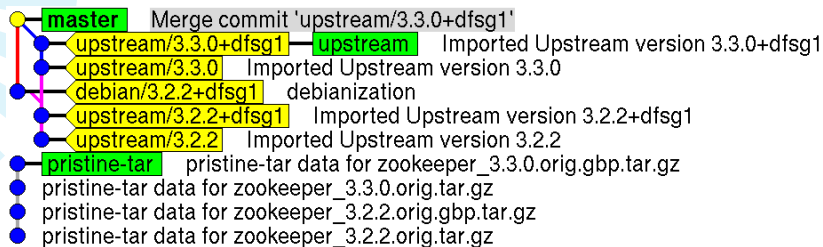
# Müll rausfiltern...

```
git-import-orig --pristine-tar --filter-pristine-tar --upstream-version=3.2.2+dfsg1 \  
--filter="*.jar" \  
--filter="Makefile.in" \  
--filter="aclocal.m4" \  
--filter="autom4te.cache" \  
--filter="compile" \  
--filter="config.guess" \  
--filter="config.sub" \  
--filter="configure" \  
--filter="ltmain.sh" \  
--filter="zookeeper-3.2.2/contrib" \  
--filter="zookeeper-3.2.2/docs" \  
--filter="zookeeper-3.2.2/recipes" \  
--filter="zookeeper-3.2.2/dist-maven" \  
--filter="zookeeper-3.2.2/src/contrib/fatjar" \  
--filter="zookeeper-3.2.2/src/c/generated" \  
--filter="zookeeper-3.2.2/src/java/main/org/apache/jute/compiler/generated/*.java" \  
--filter="zookeeper-3.2.2/src/java/generated" \  
  
../zookeeper_3.2.2.orig.tar.gz
```

gitk



## new upstream version



# Outline

Packaging mit GIT

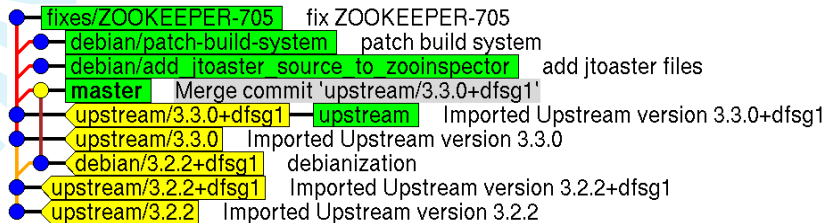
Patches in Branches



# add branches

- patches based on upstream branch

# add patch branches



## to do with patch branches

- ⊗ must be merged in the build branch
- ⊗ must be in debian/patches directory

## tnt steps in

inspired by topgit (and its shortcomings)  
requirements:

- list of branches, which are part of a patchset
- export patches to debian/patches
- apply patches to build branch
- help with upstream updates
- manage multiple patchsets in one repository  
(stable/testing/ubuntu/company internal)

# tnt design

- don't do anything special with patch branches (topgit adds .topmsg, .topdeps)
- use idea from pristine-tar to have a meta data branch in the background

# tnt implementation

- python (tried perl ...)
- git-python library (fantastic project!)
- one yaml file to hold branch references
- one message file per branch for patch description

## create patchset branch

```
git checkout upstream  
tnt init patchset_debian  
creates branch patchset_debian with yaml file specifying current  
commit as patchset root
```

# yaml file

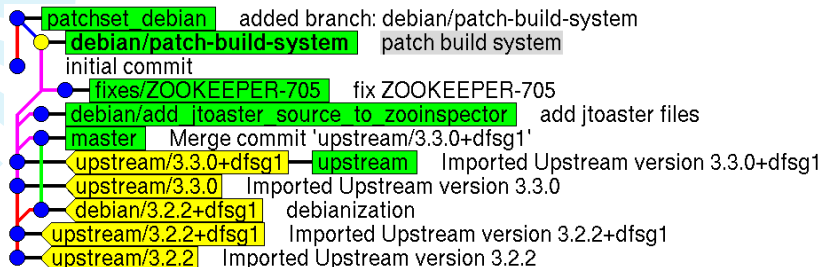
```
git show -p patchset_debian:branches
```

```
branches: {}  
root_commit: fdd5ddf140f9e7e74a42803a4a1e17998c02ddaf  
root_name: upstream
```

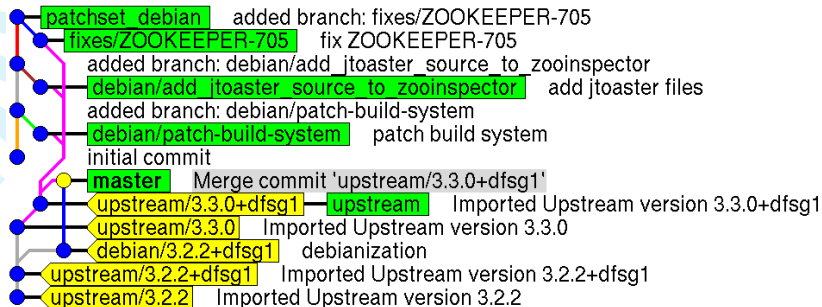
## add first patch branch

```
git checkout debian/patch-build-system  
tnt add patchset_debian
```

# first patch added



# remaining patches added



# yaml file

```
git show -p patchset_debian:branches
```

```
branches:  
debian/add_jtoaster_source_to_zooinspector:  
commit: d678bd6aca112e79885ff76ac90bb288e9d2f11a  
dependencies: []  
debian/patch-build-system:  
commit: de3eac49766644befc6e45998b4779c38f96857d  
dependencies: []  
fixes/ZOOKEEPER-705:  
commit: 91d7a547cb55e5cfe6970b078fb962c6b036e350  
dependencies: []  
root_commit: fdd5ddf140f9e7e74a42803a4a1e17998c02ddaf  
root_name: upstream
```



# export

```
git checkout -b build/3.3.0+dfsg1
tnt export patchset_debian debian/patches
vim debian/gbp.conf
commit, git-buildpackage -b
apply patches from debian/patches or merge?
```

## clone a patch set

```
git branch patchset_debian_3.3.0 patchset_debian
```



explode / implode

remove / restore references to patch branches

# TODO

- cleanup code
- implement status (to help with updates)
- implement other dependencies then ROOT
- multiple dependencies?