

Packaging for dpkg with Open Build Service

tropf

ABSTRACT

A little protocol on the steps I took to get my package into open build service to be used with ubuntu and debian.

1. Introduction

1.1. Motivation

I made `smplxmp` (<https://codeberg.org/tropf/smplxmp>), and I'd like to deliver it as package(s).

I consider packaging for debian as the biggest challenge, as it has a very large market share (and i also use it, though for more practical reasons). I created a package for `smplxmp`, mainly following the steps outlined here (<https://www.debian.org/doc/manuals/packaging-tutorial/packaging-tutorial.pdf>) and by using additional documentation I found all over the web. I will not spend too much time on explaining how to get your environment set up; this is fairly complicated and involves stuff like `gpg` etc.

1.2. Setting up `git-buildpackage`

I am using `git-buildpackage`. For an initial setup i followed this guide in the official docs (<http://honk.sigxcpu.org/projects/git-buildpackage/manual-html/gbp.import.upstream-git.html>).

To set up the created repo (that already contains the `debian/` directory) at a new machine follow these steps:

```
# clone git repo
git clone https://codeberg.org/tropf/smplxmp-deb
# configure git contact info
git config user.name "Rolf Pfeffertal"
git config user.email "floss@tropf.io"

# add upstream
git remote add upstream https://codeberg.org/tropf/smplxmp

# setup git-pbuilder
# maybe prefix with DIST=$dist
sudo git-pbuilder create
```

To incorporate upstream changes:

```
git fetch upstream
git merge $TAG
# maybe change --snapshot to --release
gbp dch --debian-branch=debian/sid --snapshot --auto debian/
# now adjust all the files
# and then check if everything builds
gbp buildpackage --git-ignore-new --git-pristine-tar --git-pristine-tar-commit --git-pristine-tar-commit

# check package
# if ok commit
git add debian/
git commit -m "..."
```

1.3. Setting up OBS

Follow this guide (<https://openbuildservice.org/help/manuals/obs-user-guide/art.obs.bg.html>).

Create a project on the openSUSE Build Service (<https://build.opensuse.org/>) and enable under "Repositories" the targets you want to build for.

Set up osc.

2. Preparing required files

After you have all your sources ready to build a package, build it locally. I'll use git buildpackage with pbuilder here:

```
gbp buildpackage --git-pristine-tar --git-pristine-tar-commit --git-pbuilder --git-pristine-tar-commit
```

From the parent directory, we now need the files

- <name>_<version>-<debversion>.dsc
- <name>_<version>.orig.tar.gz
- <name>_<version>-<debversion>.debian.tar.xz

3. Pushing package to OBS

Move these files into the directory of the package you created on obs. At this point you can test building your package. The location of the resulting files will be displayed on stdout.

```
osc build --local-package
```

After that commit & and push to obs.

```
osc addremove
osc st
# check output; if ok:
osc commit
```

> osc commit is not local; it instantly pushes to OBS. **This is not like git.**

You should now see build processes in your project on OBS.

4. Using the repositories

This is an adaptation of this (https://en.opensuse.org/openSUSE:Build_Service_Debian_builds).

> You might have to wait for quite a while after your build jobs are finished before they are turned into a repo.

4.1. Add the repo

First get the repo URL by navigating in the OBS project to your distro and click the download icon or the link "go to download repository". I will use \$URL here to refer to that repo URL. When visiting \$URL in your browser you should see a bunch of files, including files like Release or Sources.

Add the repo and its key.

› You could use `apt-key`, but that is marked as deprecated.

```
# download key (.asc is important!!)
sudo wget -O /etc/apt/trusted.gpg.d/myproject.asc $URL/Release.key
# add repo
echo "deb $URL /" | sudo tee /etc/apt/sources.list.d/myproject.list
```

4.2. Use the repo

```
sudo apt update
sudo apt install mypackage
```

5. See also

- presentation on debian packaging (<https://www.debian.org/doc/manuals/packaging-tutorial/packaging-tutorial.pdf>)
- obs introduction (<https://openbuildservice.org/help/manuals/obs-user-guide/art.obs.bg.html>)
- git-pbuilder doc (<https://wiki.debian.org/git-pbuilder>)