Cheatography

OSC - Open Build Service Command Line Client Cheat Sheet

by Lars Vogdt (kl_eisbaer) via cheatography.com/23929/cs/5379/

Create or edit a package

`osc co home:you`	Check out you(r) home project.
`cd home:you`	Change into the new directory.
`osc mkpac ctris`	Create new package named *ctris*.
`cd ctris`	Change into the new package directory.
`vi ctris.spec`	Start writing the new spec file.
`wget -q http:/ /ww w.h ack l.d hs.o rg /da ta/ dow nlo ad/ dow nlo ad.p hp ?fi le= ctr is- 0.4 2.t ar.bz2 -O ctris- 0.4 2.t ar.bz2`	Download the sources into the directory. (Note: checksum/signature validation should also be done as next step, if any of them are provided.)
`osc addremove`	Mark files to be added and/or removed from your package directory.
`osc ci -m " First checkin of the ctris packag e"`	Submit your files/changes back to the remote build instance.

Editing or creating a (new) package (meta data) in project *$PRJ* with the name *$PKG* can also be done via the command:
`osc meta pkg $PRJ $PKG -e`
everywhere in the filesystem.

To work with the sources locally, a checkout via
`osc co $PRJ $PKG`
is needed.

Branch a package

`osc branch home:lrupp ctris`	Creates a branch of the package below your home project.
`osc co home:y ou: bra nch es: hom e:l rup p/ctris`	Check out the branched package.

Branch a package (cont)

`cd home:y ou: bra nch es: hom e:l rup p/ctris`	Change into the new directory.
`vi ctris.spec`	Do changes in spec file (or other files).
`wget -q http:/ /ww w.h ack l.d hs.o rg /da ta/ dow nlo ad/ dow nlo ad.p hp ?fi le= ctr is- 0.4 2.t ar.bz2 -O ctris- 0.4 2.t ar.bz2`	Download newer source tarballs (as example).
`osc build ctris.spec`	Do a test build of the package.
`rm ctris- 0.4 1.t ar.bz2`	Cleanup before submitting, please. ;-)
`osc vc`	Add a new package changelog entry.
`osc ci -m " updated package to latest upstream versio n"`	Submit your files/changes to the remote build instance.
`osc results`	Check the build results of your changes.
`osc sr -m "Hi! I updated your package to the latest version. Have fun!"`	Create a submit request against the original project.
`osc request -M`	Check the current status of your own requests.

You might also check tools like *quilt* for fixing packages, but this is not the topic of this cheat sheet.

Maintenance updates for openSUSE follow a slightly different workflow. Please check https://en.opensuse.org/Portal:Maintenance for details.

By **Lars Vogdt** (kl_eisbaer)
cheatography.com/kl-eisbaer/
www.suse.com/

Published 31st October, 2015.
Last updated 13th May, 2016.

Cheatography

OSC - Open Build Service Command Line Client Cheat Sheet

by Lars Vogdt (kl_eisbaer) via cheatography.com/23929/cs/5379/

Building a package

`osc meta prj -e`	Either use the WebUI or edit the project configuration manually and add a build target repository.
`osc build`	Builds a package with the default values (for distribution and architecture) from the file *~/.oscrc*.
`osc build Fedora_22 x86_64`	Builds a package with given repository and architecture.
`osc lbl | less`	Shows the build log of a local build (pipe into pager to allow scrolling and searching).
`osc chroot`	Jump into the chroot environment of the current local build. Might be useful for debugging and/or fixing a package.
`osc ci -m " - fixed package build for Fedora "`	Submit your files/changes to the remote build instance.
`osc results`	Shows the build results of a package or project.
`osc prjresults`	Shows project-wide build results.

Building a package locally saves time, as the public instances need to find a free schedule for building, while the local machine can start the build immediately. As the build is done in a chroot environment, the OS installation on the local machine will not be affected.

Please remember that the public build instances do neither allow network access nor root permissions during build.

Collaborating on packages

`osc request list home:lrupp`	Check for (open) submit requests against a repository.
`osc request show -d $ID`	Deeply inspect submit request with id $ID (show diff).
`osc request accept $ID -m " Thank you for your contri bution :-)"`	Accept a pending request.
`osc request decline $ID -m " Sorry, but you forgot to add missing files"`	Decline a pending request.
`osc request supersede -m "He did it better than me - use his submit request please " $ID $SUPER SED - ING_ID`	Superseede a pending submit request with another one.
`osc branch -N -M openSU - SE: Bac kpo rts :SL E-1 - 2:U pda te/ glibc7`	Create a branch pointing to a not yet existing package for a project and package in maintenance mode like the package hub project.

Working with meta informatoin

`osc meta prj $PRJ`	Show meta informatoin about a project.
`osc meta pkg $PRJ $PKG`	Show meta information about a package.
`osc meta user $USER`	Show information about a user.
`osc meta prjconf $PRJ`	Show meta information about a project.
`osc update pac - met afr omspec`	Update package meta data with metadata taken from spec file.

osc comes with a very useful `--help` option and man page. Please refer to this material if you want to get more information.

- ✔ https://en.opensuse.org/openSUSE:OSC
- ✔ https://github.com/openSUSE/osc

By **Lars Vogdt** (kl_eisbaer)
cheatography.com/kl-eisbaer/
www.suse.com/

Published 31st October, 2015.
Last updated 13th May, 2016.