RPM Package Build by Open Build Service Software

This document provides steps of RPM package build by Open Build Service software.

The example here shows how to build Gbs and Mic rpm packages successfully.

You could build other rpm packages when they are necessary by referring to this example.

# Prerequisites

## Make sure osc package has been installed in the linux system , and the CONFIG please Refer to Appendix A

* 1. Make sure you have privilege to login the Open Build Service by

<https://build.otctools.jf.intel.com/>. If you don’t have privilege please contact the manager to apply.

# Notes

## Gbs and Mic should be located in the same obs project

## This obs project should be located in the home project of OBS.

## This project should have only one openSUSE12.03’s repo to keep testing environment clean.

## Add the testing folks to test this project. The specific way please follow the pic in 3.2.

## When building MIC and GBS, it’s needed to copy the packages of tools/devel in the project of OBS to the project

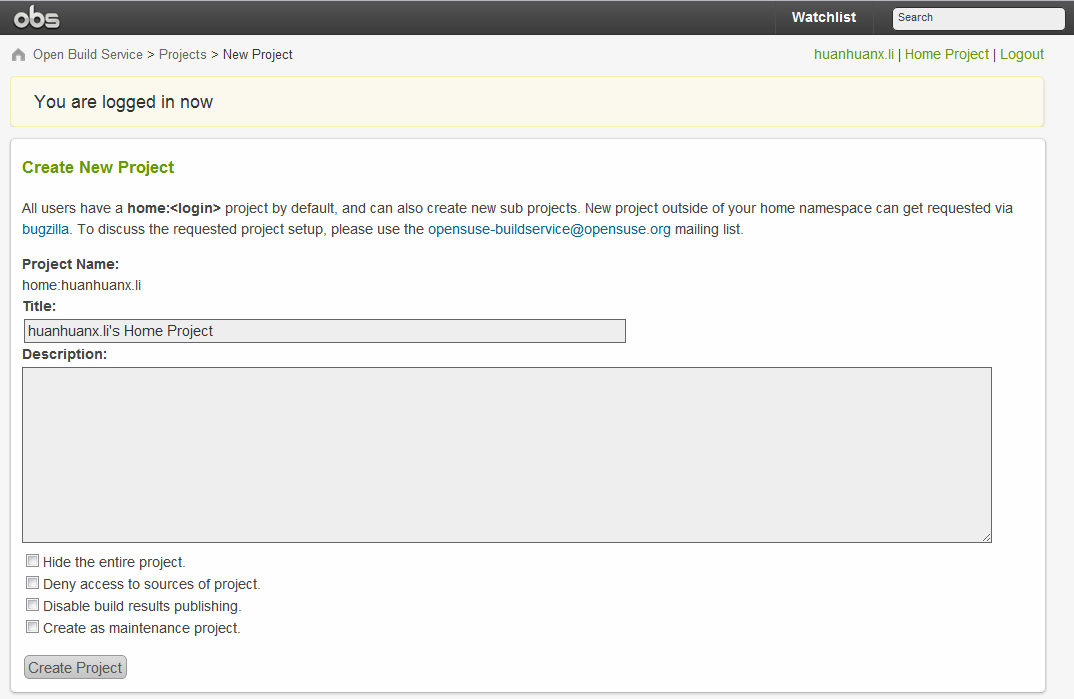
# Steps of building a repository on OBS

## Create Project

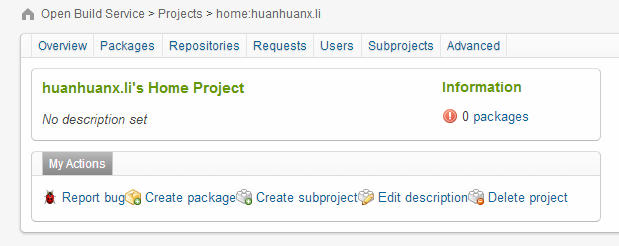
If you don’t have home project, please follow 3.1.1 to create your home project first.

Else please just follow 3.1.2 to create a new sub-project directly.

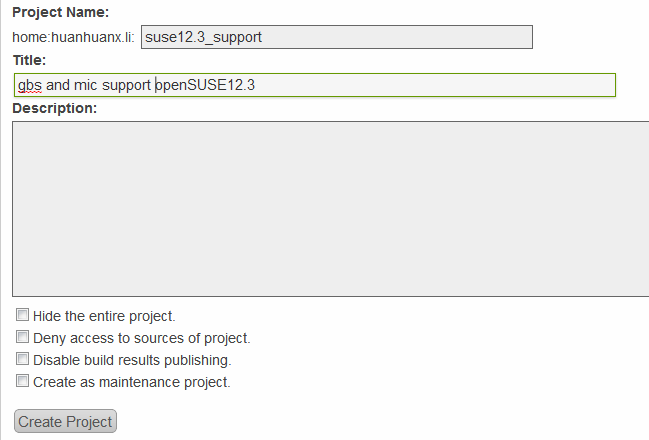
### Create New Project as the pic below, fill the blanks, then Create Project.

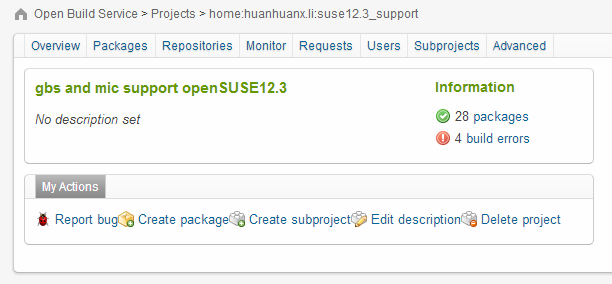


### Click “Create Subproject” in the home project as the pic below



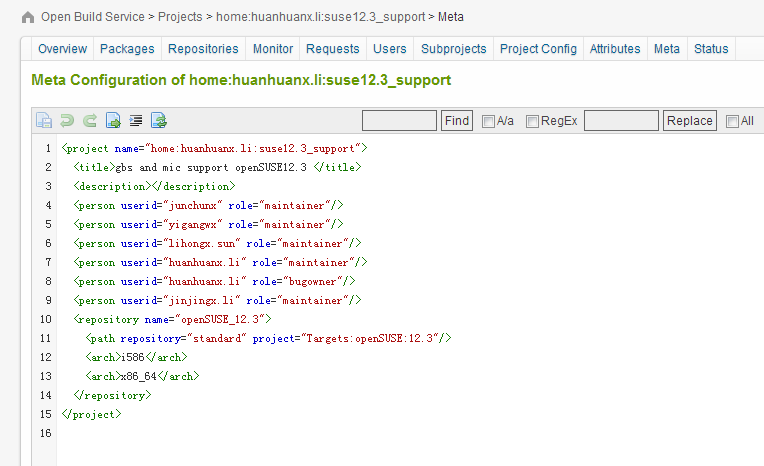
### Fill the blanks of Create New Subproject, then Create Project and generate the suse12.3\_support project as the second pic.





## Add repository of openSUSE12.3

## As show in the pic, find Meta under your home project, the repository part in HTML is the way to add a repo; and person part is the way to add a membership, then choose SAVE .



## Operation about create packages

### Please refer to Appendix A to learn :Get home project、 Get Gbs and Mic source code packages from Tools:Devel project from obs.

### Create packages

The code below describes that you should do 3.3.1 step first, then copy source code from Tools:Devel directory to your home project directory locally.

You do add and commit package under your home project directory locally, it will be committed to your home project in obs server at once, and you could watch obs building it automatically.

The succeeded result as shown in 3.4 pic is what we want, it means rpm package has been built successfully.

home:hhx.li:suse12.3\_support $ osc mkpac gbs

home:hhx.li:suse12.3\_support/gbs$

cphome/Tools\:Devel/gbs/\* .

home:hhx.li:suse12.3\_support/gbs $ osc st

home:hhx.li:suse12.3\_support/gbs $ osc add \*

home:hhx.li:suse12.3\_support/gbs $ osc ci –m ‘add gbs’

home:hhx.li:suse12.3\_support $ osc mkpac mic

home:hhx.li:suse12.3\_support/mic$

cphome/Tools\:Devel/mic/\* .cp ~/obs-project/Tools\:Devel ./

home:hhx.li:suse12.3\_support/mics$ osc st

home:hhx.li:suse12.3\_support/mic $ osc add \*

home:hhx.li:suse12.3\_support/mic $ osc ci –m ‘add gbs’

### Check the target in the new repository

Show as the pic below, from the new repo we could see the build results of two rpm packages are succeeded, which means we could download them or install them by this repo now.

If the result isn’t “succeeded” but others as failed、broken、unresolvable and so on, we should refer to Legend and find out the unsuccessful reasons, maybe now we’d better review what Notes 2.5 advices us, and retry it according to 3.3.



Stop here, we have built the gbs and mic rpm packages successfully, and the next what we need to do is install these rpm packages in openSUSE12.3 by Itest or do it step by step manually in linux system. At that time we will meet new problems and as well conceive handle methods to fix bugs.

# Appendix I

## CONFIG

$ vim .oscrc

[general]  
apiurl = https://api.opensuse.org  
  
[https://api.opensuse.org]  
use=username  
pass=xxxxxxxx  
aliases=suse  
trusted\_prj=openSUSE:12.2 CentOS:CentOS-6  
  
[https://api.otctools.jf.intel.com]  
sslcertck=0  
aliases=otctools  
user= username  
pass=xxxxxxx  
trusted\_prj=Targets:Ubuntu:12.04 Targets:Ubuntu:11.10 Tools:Devel Tools openSUSE.org:Fedora:17 Targets:openSUSE:12.1 Targets:Ubuntu:12.10 Targets:Fedora:17

## Get home project from obs

$ osc –A otctools co home:huanhuanx.li:suse12.3\_support

$ cd home\:hhx.li\:suse12.3\_support

home:hhx.li:suse12.3\_support $ ls

## Get Gbs and Mic source code packages from Tools:Devel project in obs

$ osc –A otctools co Tools:Devel gbs

$ osc –A otctools co Tools:Devel mic

Tools:Devel $ ls

gbs mic

Tools:Devel/gbs/ $ ls

gbs\_0.15.tar.gz gbs.changes gbs.dsc gbs.spec Makefile

## Some relational links

### English version of official OSC comment document

<http://en.opensuse.org/openSUSE:OSC>

### Chinese version of official OSC comment document

<http://cn.opensuse.org/Build_Service/CLI>

### OpenSUSE Build Service Cheat Sheet

<http://en.opensuse.org/images/d/df/Obs-cheat-sheet.pdf>

# Appendix II

## 5.1 Issues

### 5.1.1 openSUSE 12.3

* Qemu-arm-static unresolvable glib2-devel = 2.34.3 needed by glib2-devel static

Disable glib2 or delete glib2 then qemu-arm-static will success build

### 5.1.2 Ubuntu 13.04

* Build result like:

Gbs no provides python-support

So we should check on a ubuntu13.04 VM

$ apt-cache search python-support

paraview-python - Parallel Visualization Application. python-support

python-support - automated rebuilding support for Python modules

we can find this package python-support from repo, so we need to fix this. Just to tell system admin to add this package.