

Openbuildservice

[imx6](#), [obs](#), [compile](#), [worker](#), [packages](#), [drivers](#)

Introduction

Building, maintaining and publishing packages can become a real lot of effort when their number, or the number of supported systems increase. The Open Build Service is a fully-automated system to automatically build and publish packages for several distributions including openSUSE, Fedora, Debian and Archlinux. In addition it provides a nice Web UI to manage packages, projects, build configurations as well as to show sources and buildlogs. Furthermore there is a command-line client available that works like a version-control-system.

The system is currently hosted here: <https://obs.solid-build.xyz/> The repositories are located at: <https://obs.solid-build.xyz:82/> However for production the mirror <https://repo.solid-build.xyz/> should be used!

Setting up the build target

In every home-project there is a repositories tab. Repositories are basically a link to the binaries of another project on the OBS. On that tab repositories can be added. It is important not to take the easy suggestions listed such as Debian 7.0, openSUSE. Instead, use the “pick one via advanced interface” link to get an advanced dialog. There, the Project that one wants to build against, as well as the specific repository in that project need to be specified.

There are currently the following repositories(Projects) providing supported distributions as build-targets:

```
Distributions:Debian_Sid/main
Distributions:Debian_Jessie/main
Distributions:Debian_Wheezy/standard
Distributions:Fedora-23/everything
Distributions:Fedora-24/everything
openSUSE.org:openSUSE:13.2/ports
```

Additional distributions can be added on request.

Sometimes one requires additional packages from another project, or perhaps distribution updates. To accomplish that, additional repositories can be included in a project's repository. Say your project is home:Admin, and you got a repository called Fedora_20 (that points at Distributions:Fedora_20/standard), and you want to add fedora update. Then this is what you have to do: Go to the repositories tab of home:Admin Under Fedora_20, follow link “Edit repository”, then “Add additional path to this repository”. In the project field belongs “Distributions:Fedora_20”, and in the repository list box below, choose “update”.

Now the same can be done for any other projects. An alternative method is to edit the projects metadatafile under the Advanced→Meta tab. For this method it is best to look at another projects metadata and modify it accordingly.

Debian OBS packaging work-flow

The first step is to create a new package inside an OBS project. OBS provides a version-control-system for it and a gui to create/modify files.

Raw Development

To build a debian package on the OBS it is required to upload a .dsc and all the source tarballs to OBS. All of these can be recreated by running

```
dpkg-source -b .
```

in the source directory. As an alternative, git-buildpackage will create them too, after a successful build.

git-buildpackage

If you like using git, you can work on your applications and packaging in a git repository. The OBS can then automatically fetch sources from that repo to perform builds. As a rule of thumb, if

```
gbp buildpackage --git-ignore-branch
```

works without any additional arguments for you, the OBS should be okay too.

To make the OBS actually download the sources, a new file alled “_service” has to be created, and uploaded to the OBS:

That's what Source-Services were invented for. They are run server-side on the OBS and download sources from remote locations and Version-Control-Systems. For debian packages, a wrapper around git-buildpackage is used that generates the .orig.tar.gz, .debian.tar.gz and the .dsc directly from a git repository. The following is a sample source-service. It has to be saved as _service inside the package (OBS package).

```
<services>
<service name="gbp">
<param name="url">https://github.com/dv1/irqbalanced.git</param>
<param name="revision">master</param>
</service>
</services>
```

Other

There are a few other source-services available for the OBS. Some to clone repos, some to extract individual files form archives, ... There are plenty of ways to manage your sources, just pick what suits you best!

Commandline Client

The Web-Interface to OBS can quickly become a bottleneck in the development process. Thats why there is a commandline-client called "osc" that works like a version control system. With it you can work on your obs projects and packages on your local filesystem, copy packages, branch packages, trigger rebuilds or source-service runs. It is best to refer to the openSUSE wiki page for detailed informations: <https://en.opensuse.org/openSUSE:OSC>

Useful Links

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

https://en.opensuse.org/openSUSE:Packaging_guidelines

From:

<https://wiki.solid-run.com/> - Wiki | SolidRun

Permanent link:

<https://wiki.solid-run.com/doku.php?id=products:imx6:software:development:obs>

Last update: **2016/06/23 02:56**

