Best Practice Guide

Best Practice Guide: Open Build Service

by Adrian Schröter, Björn Geuken, and Moisés Déniz Alemán

Publication Date: 01/23/2019

SUSE LLC
10 Canal Park Drive
Suite 200
Cambridge MA 02141
USA
https://www.suse.com/documentation

Contents

About this Guide

This guide is part of the Open Build Service documentation. These books are considered to contain only reviewed content, establishing the reference documentation of OBS.

This guide does not focus on a specific OBS version. It is also not a replacement of the documentation inside of the openSUSE Wiki (https://en.opensuse.org/Portal:Build_Service). However, content from the wiki may be included in these books in a consolidated form.

1 Available Documentation

The following documentation is available for OBS:

Book "Administrator Guide"

This guide offers information about the initial setup and maintenance for running Open Build Service instances.

Article "Beginner's Guide"

This guide describes basic workflows for working with packages on Open Build Service. This includes checking out a package from an upstream project, creating patches, branching a repository, and more.

Best Practice Guide

This guide offers step-by-step instructions for the most common features of the Open Build Service and the openSUSE Build Service.

Book "Reference Guide"

This guide covers ideas and motivations, concepts and processes of the Open Build Service and also covers administration topics.

Book "User Guide"

This guide is intended for users and developers who want to dig deeper into Open Build Service. It contains information on backgrounds, setting up your computer for working with OBS, and usage scenarios.

2 Feedback

Several feedback channels are available:

Bugs and Enhancement Requests

Help for openSUSE is provided by the community. Refer to https://en.opensuse.org/Portal:Support for more information.

Bug Reports

To report bugs for Open Build Service, go to https://bugzilla.opensuse.org/, log in, and click *New*.

Mail

For feedback on the documentation of this product, you can also send a mail to `doc-team@suse.com`. Make sure to include the document title, the product version and the publication date of the documentation. To report errors or suggest enhancements, provide a concise description of the problem and refer to the respective section number and page (or URL).

3 Documentation Conventions

The following notices and typographical conventions are used in this documentation:

- `/etc/passwd`: directory names and file names
- *`PLACEHOLDER`*: replace *`PLACEHOLDER`* with the actual value
- `PATH`: the environment variable PATH
- **`ls`**, `--help`: commands, options, and parameters
- `user`: users or groups
- `package name` : name of a package
- Alt, Alt–F1: a key to press or a key combination; keys are shown in uppercase as on a keyboard
- *File*, *File* › *Save As*: menu items, buttons
- *Dancing Penguins* (Chapter *Penguins*, ↑Another Manual): This is a reference to a chapter in another manual.

- Commands that must be run with `root` privileges. Often you can also prefix these commands with the **`sudo`** command to run them as non-privileged user.

```
root # command
geeko > sudo command
```

- Commands that can be run by non-privileged users.

```
geeko > command
```

- Notices

Warning: Warning Notice

Vital information you must be aware of before proceeding. Warns you about security issues, potential loss of data, damage to hardware, or physical hazards.

Important: Important Notice

Important information you should be aware of before proceeding.

Note: Note Notice

Additional information, for example about differences in software versions.

Tip: Tip Notice

Helpful information, like a guideline or a piece of practical advice.

4 Contributing to the Documentation

The OBS documentation is written by the community. And you can help too!

Especially as an advanced user or an administrator of OBS, there will be many topics where you can pitch in even if your English is not the most polished. Conversely, if you are not very experienced with OBS but your English is good: We rely on community editors to improve the language.

This guide is written in DocBook XML which can be converted to HTML or PDF documentation.

To clone the source of this guide, use Git:

```
git clone https://github.com/openSUSE/obs-docu.git
```

To learn how to validate and generate the OBS documentation, see the file `README`.

To submit changes, use GitHub pull requests:

1. Fork your own copy of the repository.
2. Commit your changes into the forked repository.
3. Create a pull request. This can be done at https://github.com/openSUSE/obs-docu.

It is even possible to host instance-specific content in the official Git repository, but it needs to be tagged correctly. For example, parts of this documentation are tagged as `<para os="opensuse">`. In this case, the paragraph will only become visible when creating the openSUSE version of a guide.

1 Using the OBS Web UI

This chapter explains and shows how you could use OBS Web UI. We will show and use OBS Web UI based on http://build.opensuse.org. You need to make an account first to follow this chapter contents.

1.1 Homepage and Login

Open a browser and navigate to https://build.opensuse.org

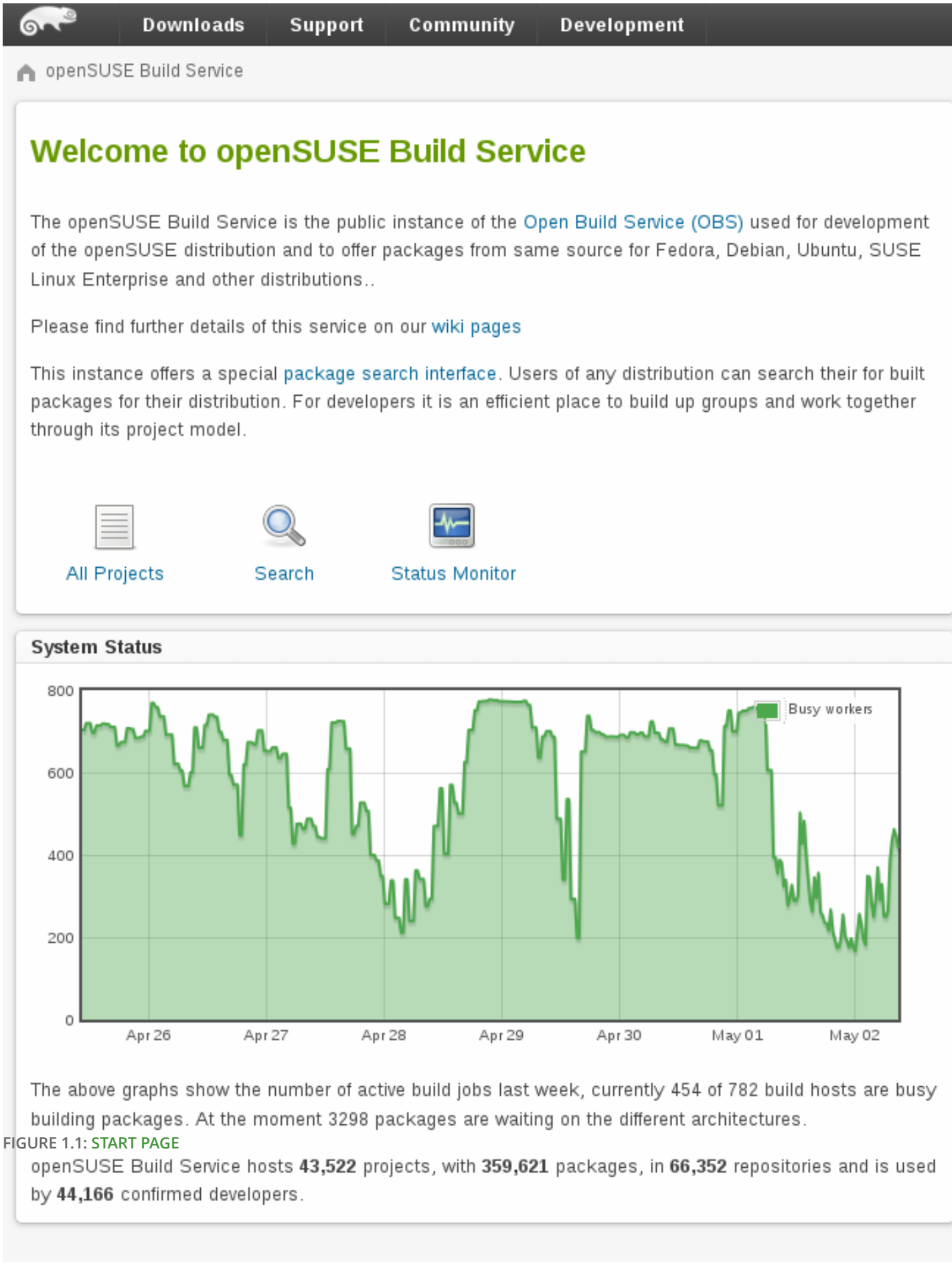

FIGURE 1.1: START PAGE

To proceed, you'll need to log in and authenticate with your username and your password. Click on Login and enter the data in the upper right corner.

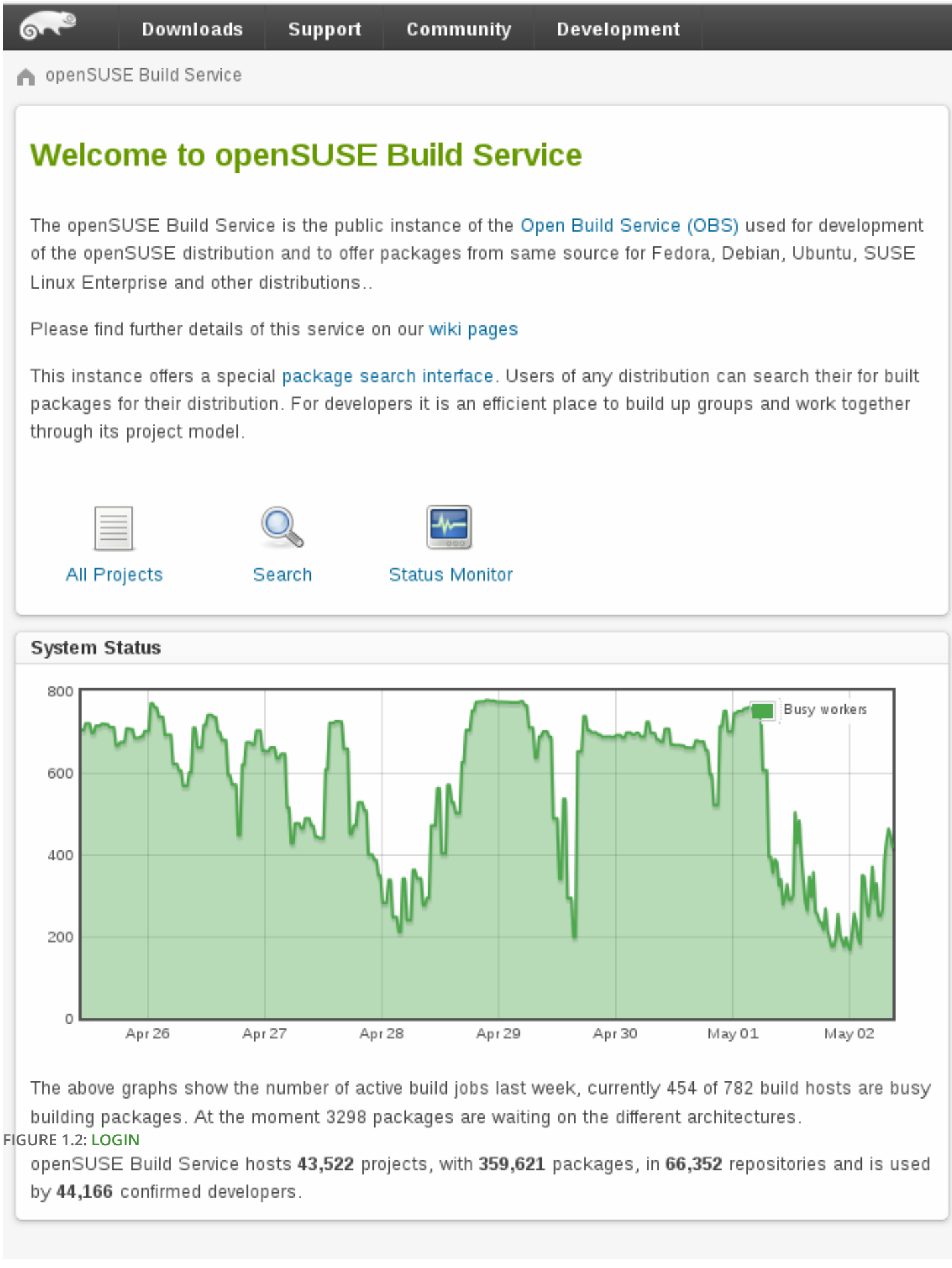

FIGURE 1.2: LOGIN

After successful authentication, you'll end up on the start page again - with new options visible. We'll go through most of them in detail, but first lets create your home: in the next step.

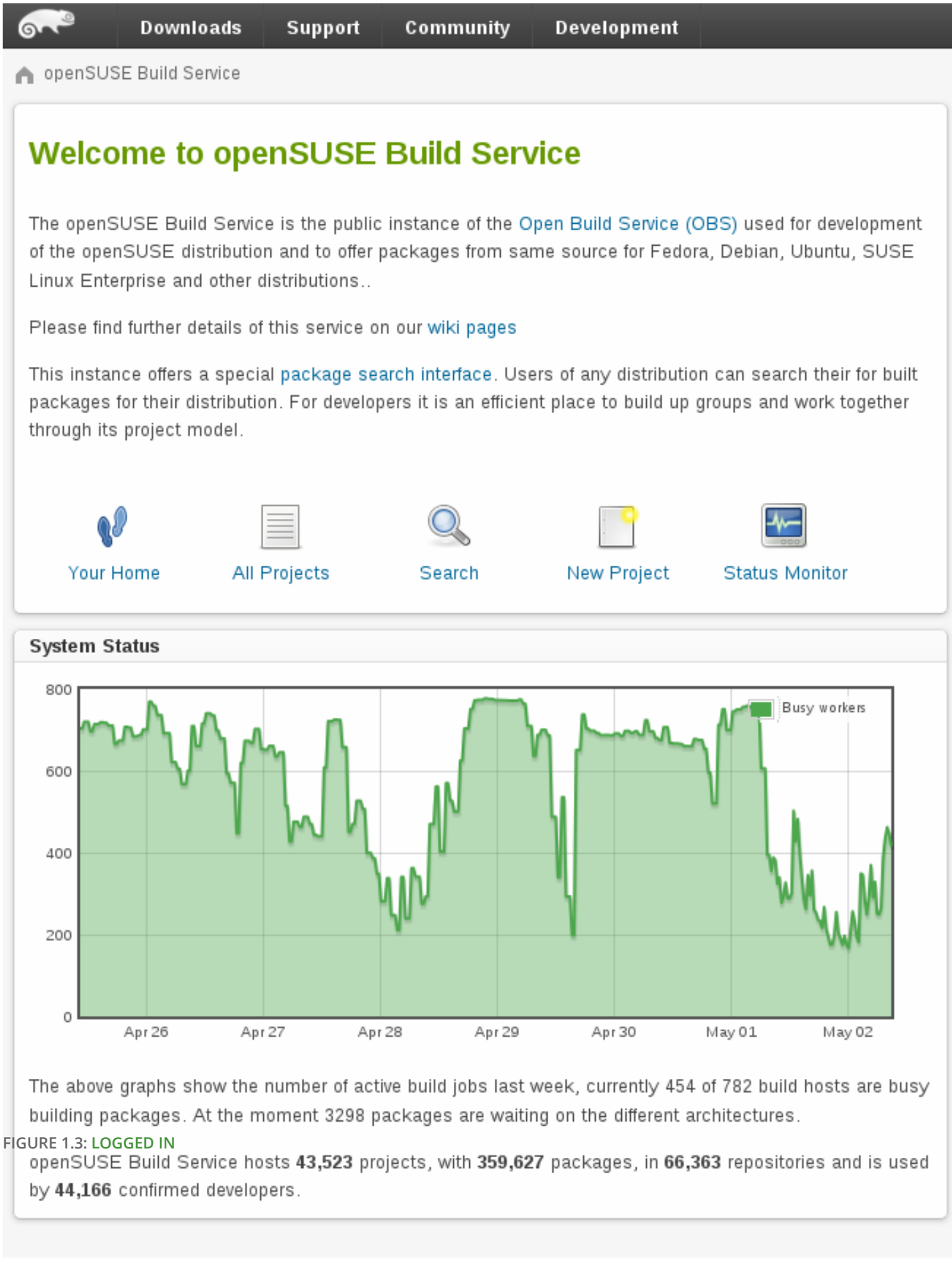

FIGURE 1.3: LOGGED IN

1.2 Home Project

Every user has a home project (home:[userid]) where they have write access by default. This is a personal workspace where you can experiment and play. Click on the link "Home Project" at the upper right to get to your home project.

1.2.1 The Project Page

Your home project will be empty for now, but you can add packages containing sources/build recipes and projects which are containers for the build targets. As you can see, you're the default maintainer which grants you full write access to this project. You're also the bug owner of your project.

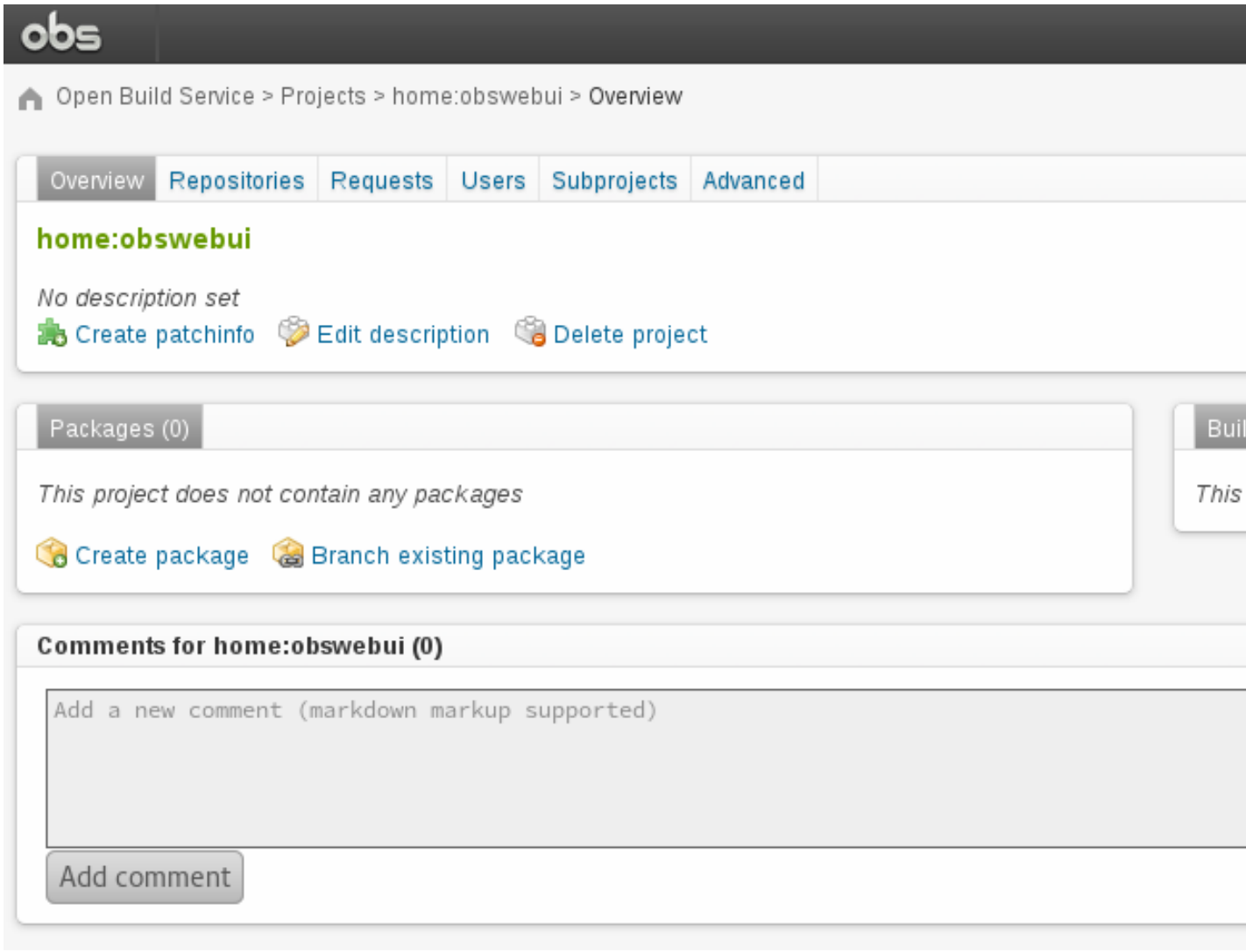

FIGURE 1.4: PROJECT PAGE

1.2.2 Changing a project's title and description

On every project page you will find a "Edit description" link. This link will lead you to a place were you can review and change your project's title and description. Click on the "Update project" button to save.

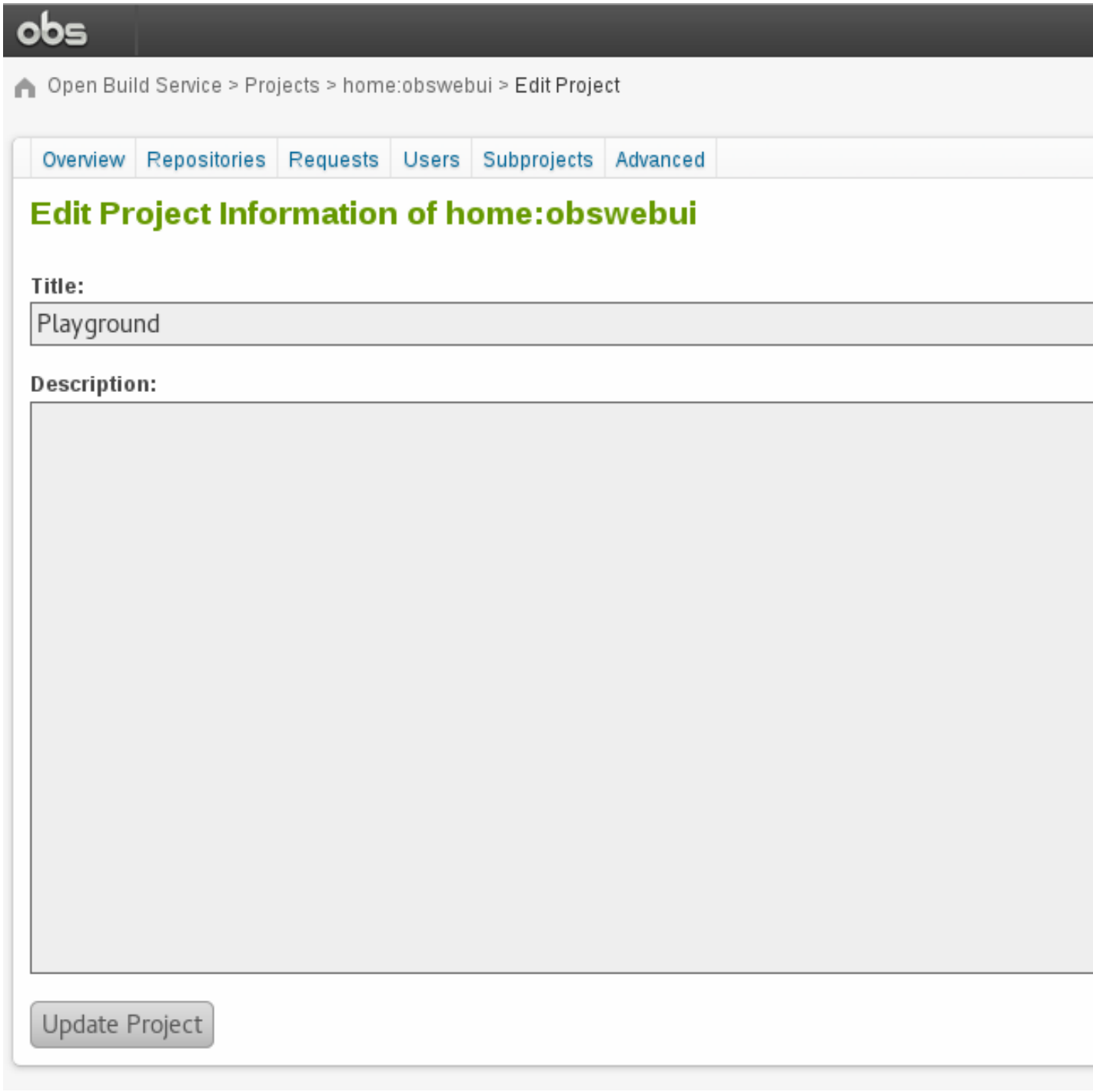

FIGURE 1.5: UPDATING PROJECT DESCRIPTION

1.2.3 Creating Subprojects to a Project

Subprojects are projects that are part of another projects namespace. Subprojects are an easy way to organize multiple projects. On the "Subprojects" tab you can find a list subprojects that belong to a project. To create a new subproject click on the "New subproject" link, fill in the form and press the "Create project" button.

Note

Maintainers of upper projects can always modify the subprojects. Apart from that all projects are separated and have no influence on each other.

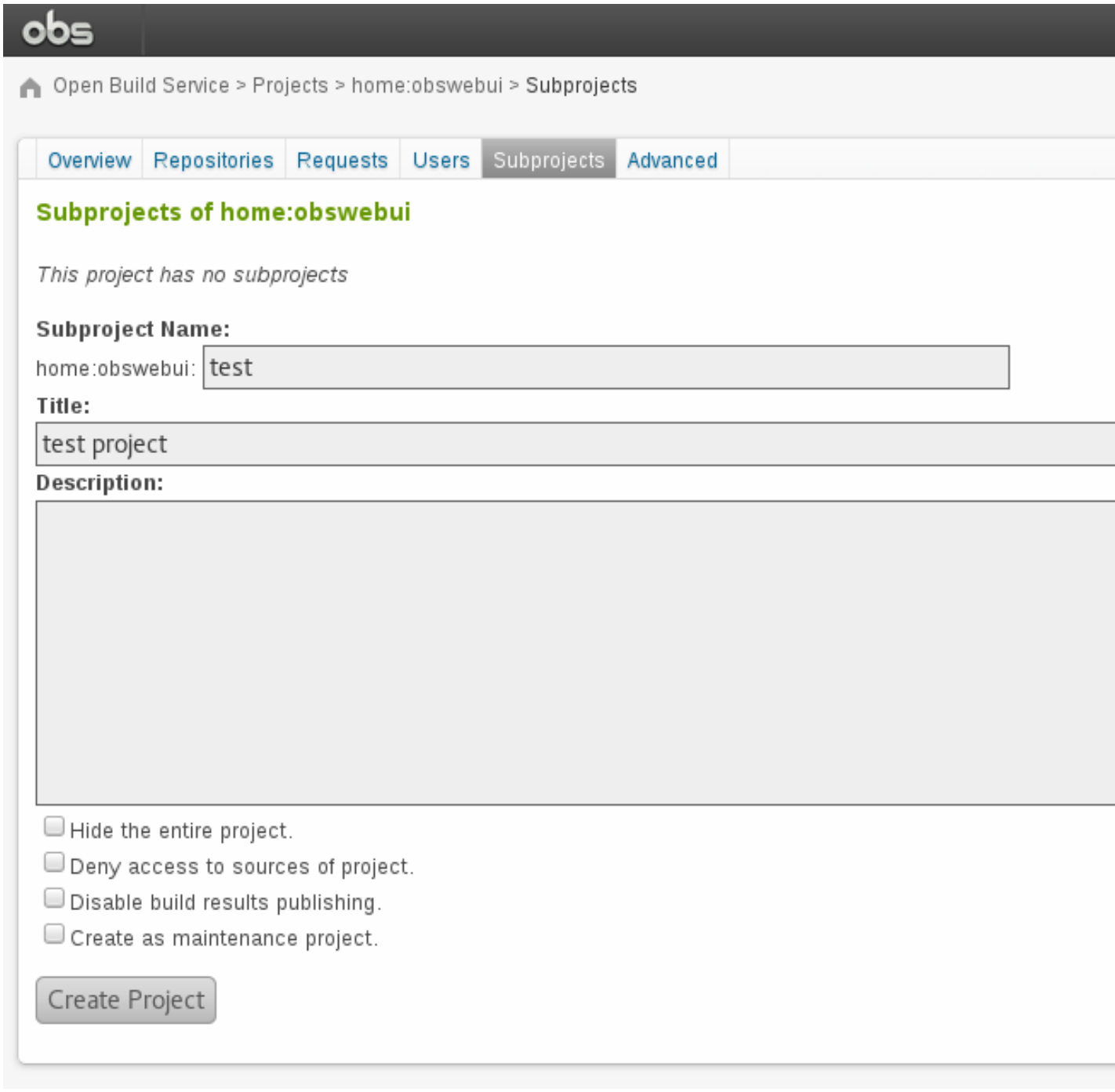

FIGURE 1.6: CREATING SUBPROJECTS

1.3 My Projects, Server Status

For now, let's leave your home for a bit and explore the build service. Click on "My Projects" on the left at the bottom. This opens a page listing your watched projects and your involvements in projects or packages.

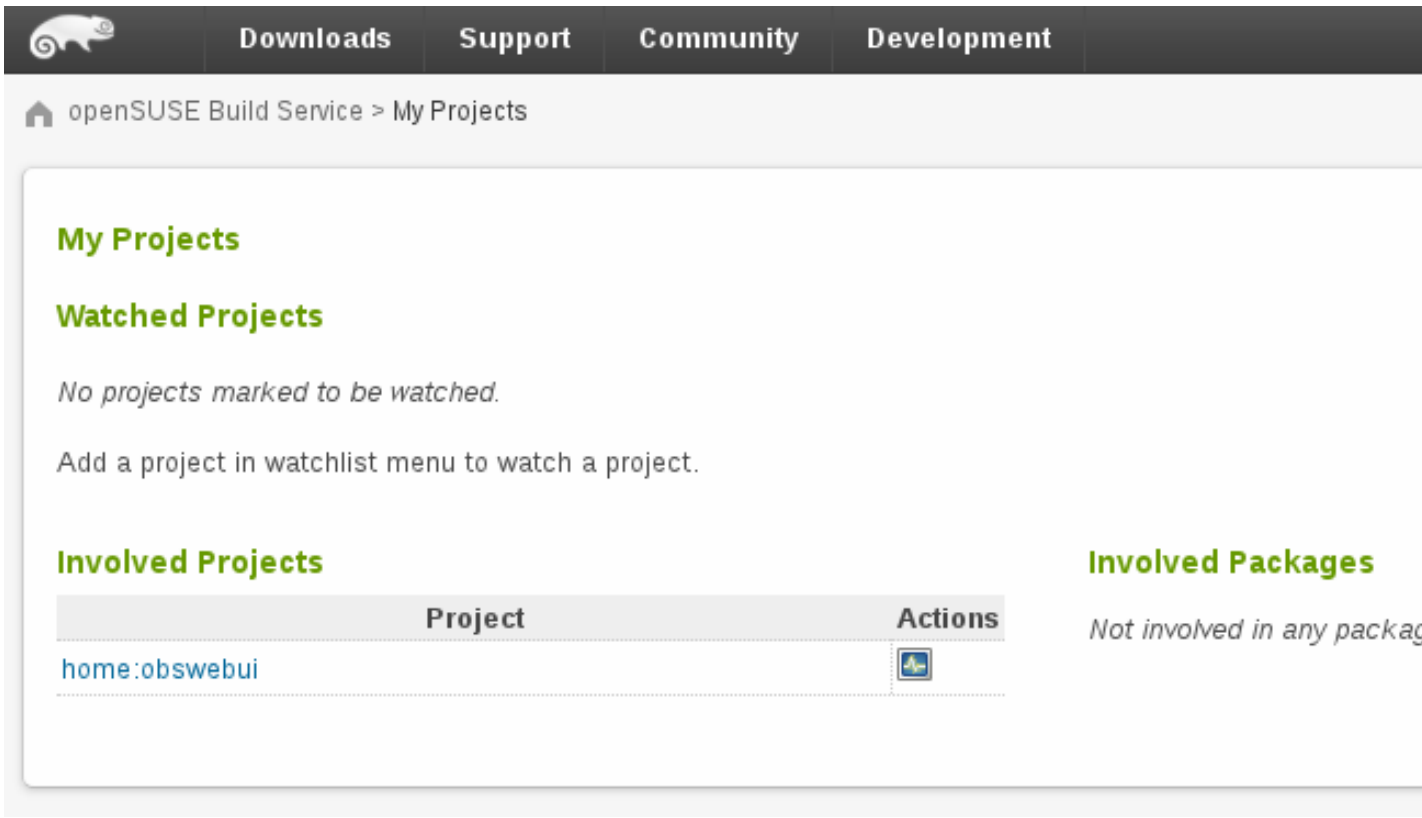

FIGURE 1.7: MY PROJECTS

Now, let's visit the main monitor page by clicking on "Status Monitor". You see here the status of the services, some graphs and graphics are showing the currently running and completed jobs and the overall load.

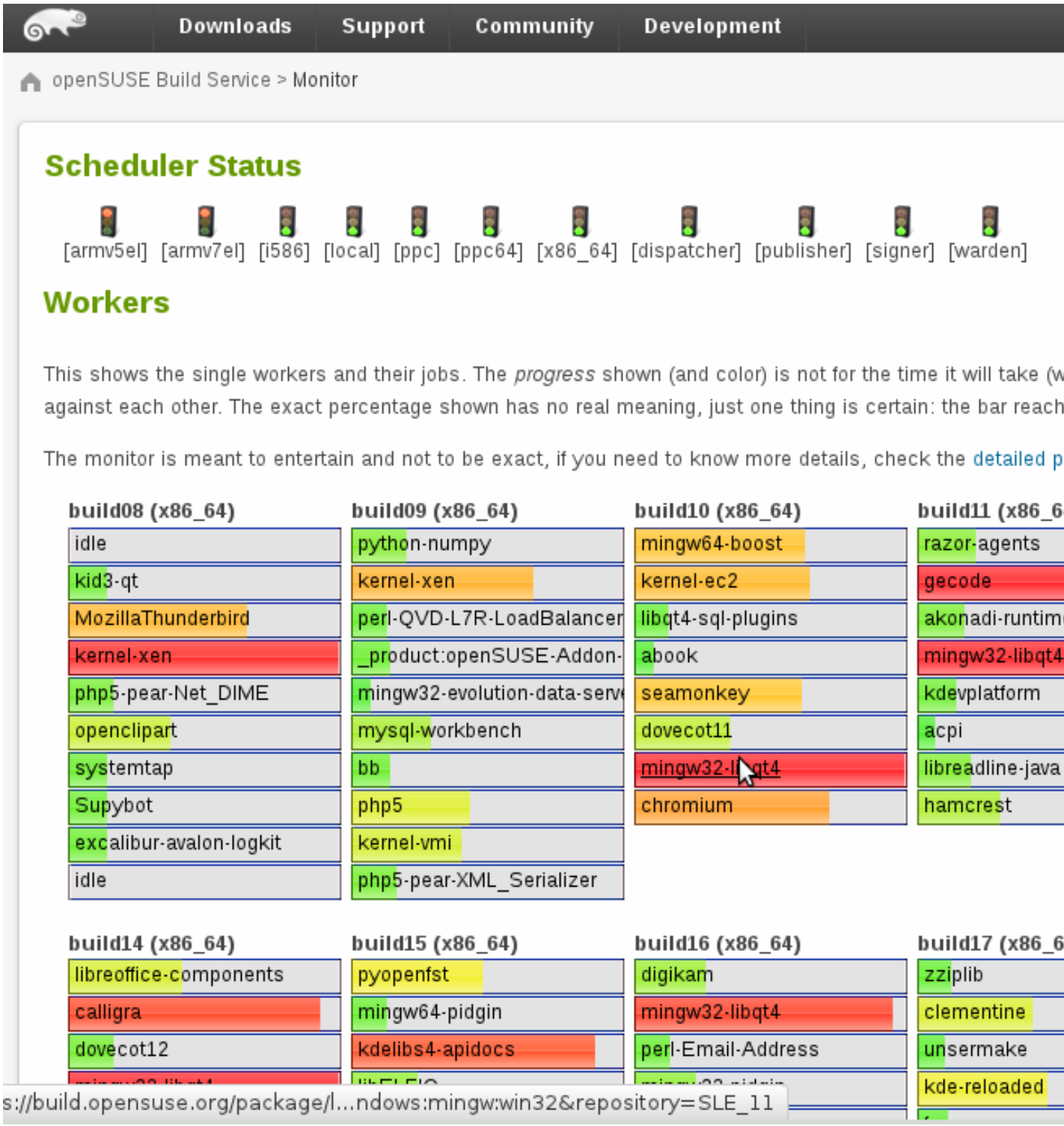

FIGURE 1.8: STATUS MONITOR

1.4 Create a link to a package in your home:

We'll show you how you can log in and use the web interface hosted at build.opensuse.org. This includes login, adding a link to a package in your personal workspace (home:) and how to build that package by adding a repository. First, let's enter "My Projects" by clicking on the link at the bottom left.

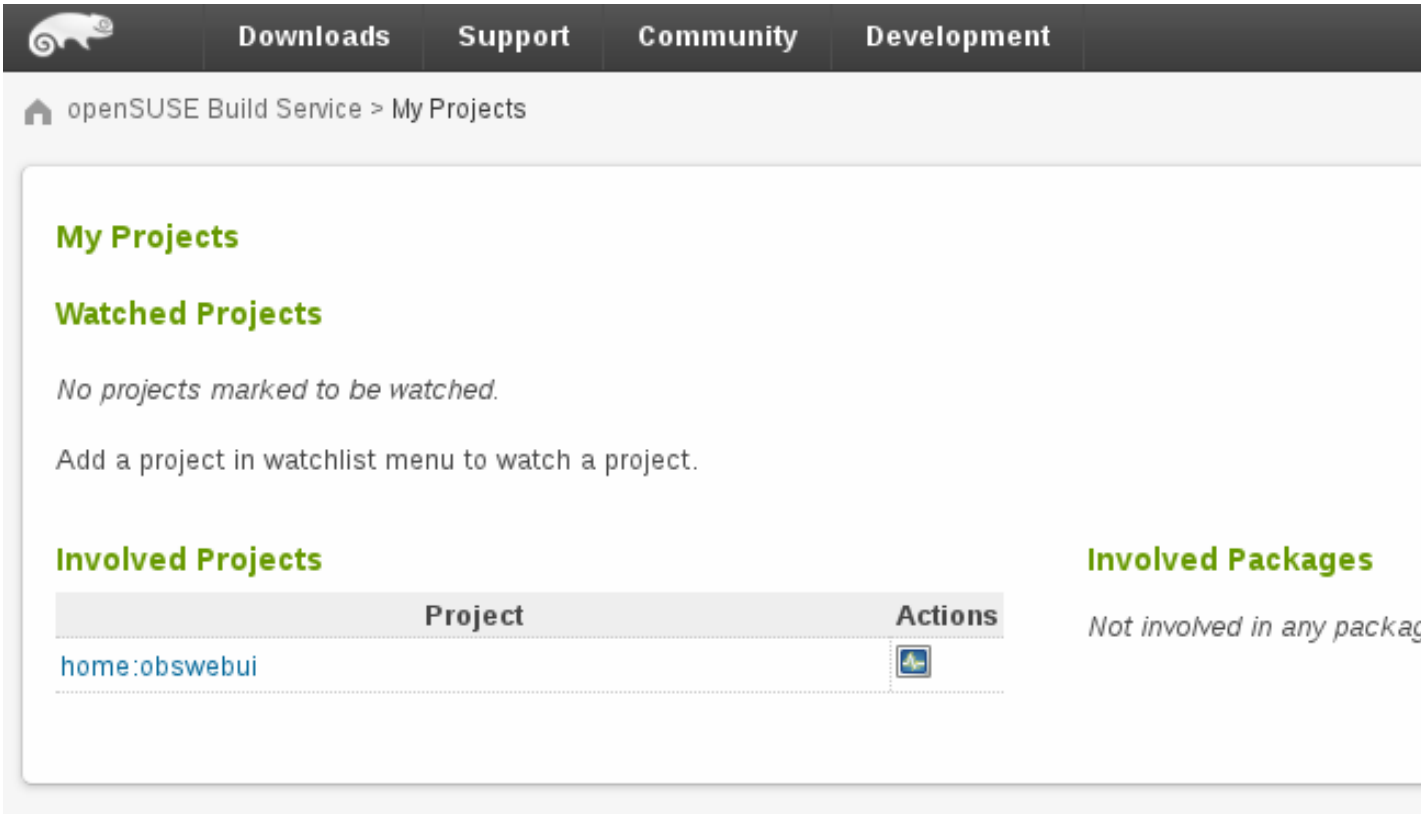

FIGURE 1.9: MY PROJECTS

Now let's create a link to a package and add a repository to build against. A link is basically a pointer to sources of an already existing package. By "repository" we mean container of built binary packages like Debian_8 or openSUSE_13.2. Let's follow these steps:

1. Add link to the existing package.
2. Add repository.
3. Observe the build on the monitor page.
4. Look at package's page.

1.4.1 Add Link to Existing Package

Right below packages, there's "Branch Package from other Project" .

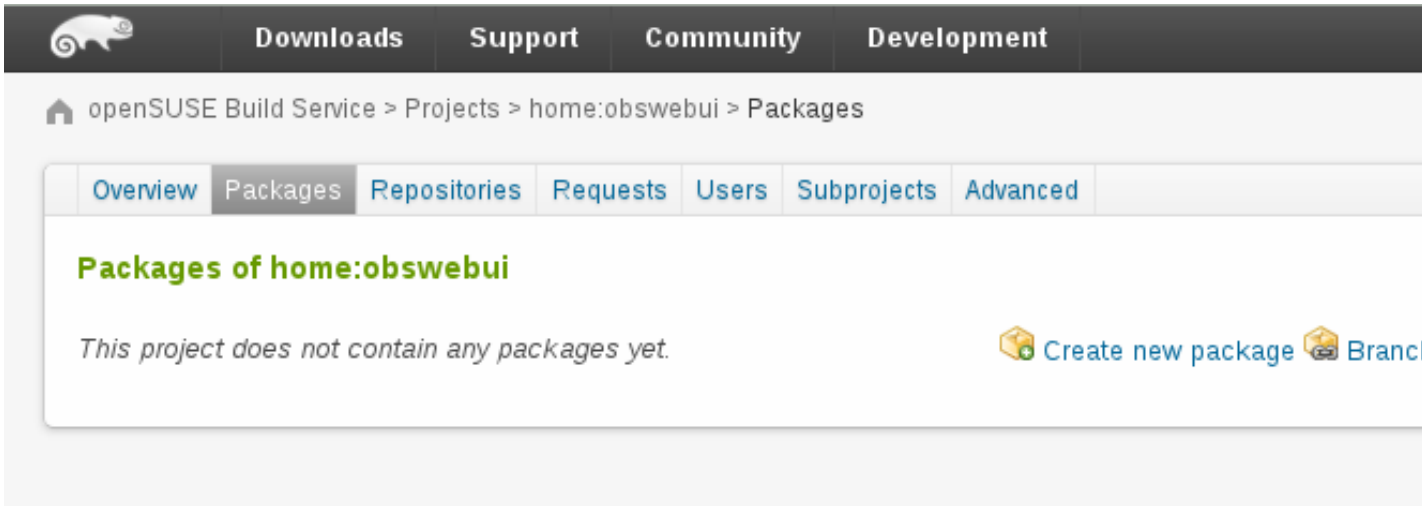

FIGURE 1.10: BRANCH PACKAGE

Open that page and enter for

```
Name of original project:
Apache
```

and for

```
Name of package in original project:
flood
```

- we'll leave "Name of linked package in target project" empty. This is shown on the next picture:

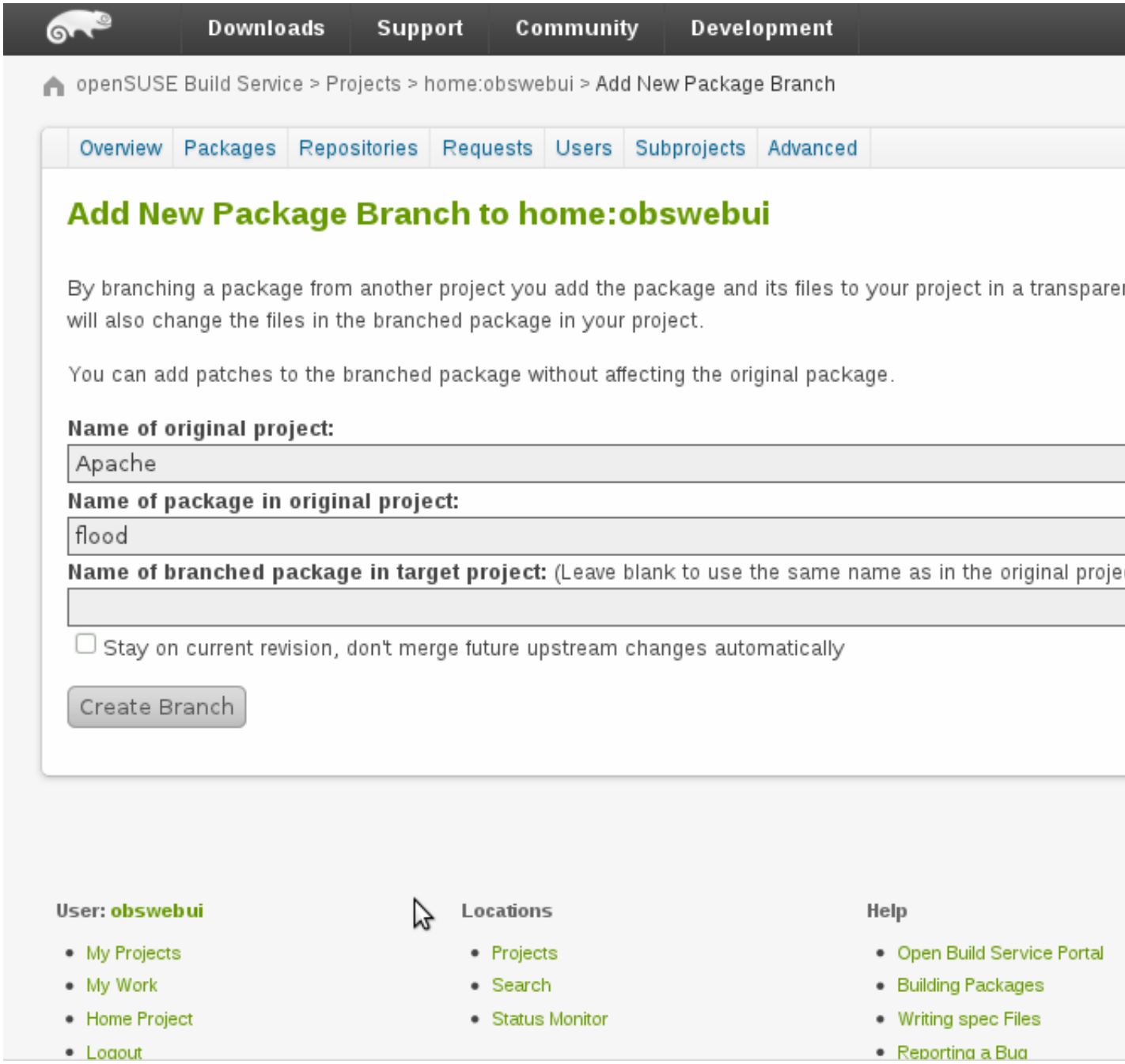

FIGURE 1.11: APACHE FLOOD BRANCH

Proceed with "Create Branch" and you'll be redirected to your home again. You'll see a new package "flood" and a notice about the branch being added.

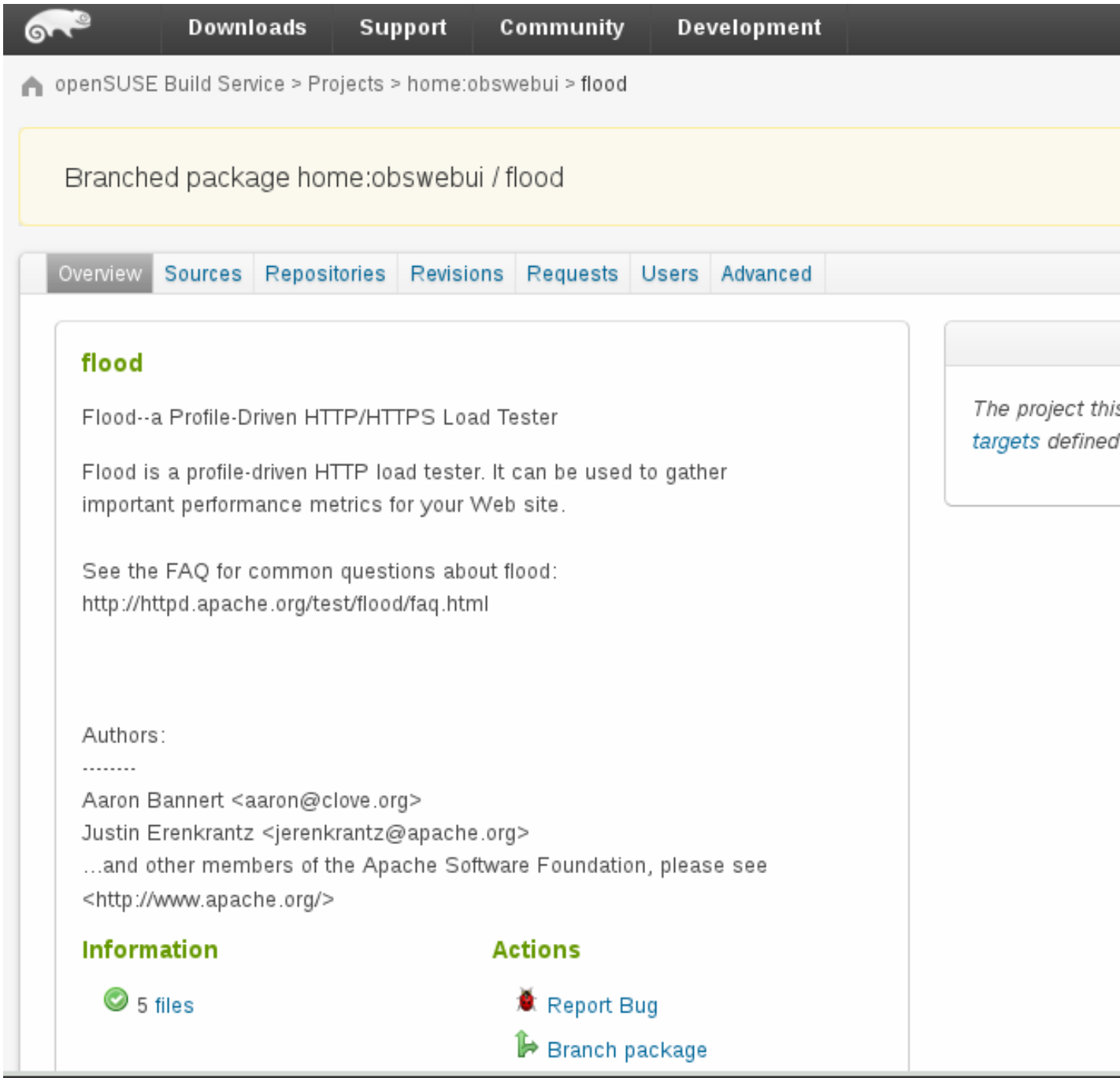

FIGURE 1.12: BRANCHED PACKAGE

Wonderful, we've added a pointer to the sources! Now we need to add a repository, so the builder knows the target-distribution to build packages for. How to add a repository to a project is documented at *Section 1.6.1, "Adding a repository"*.

1.4.2 Package Page, Build Log and Project Monitor Page

Next, it is time to explore the Monitor page, the package detail page and the build log. Just click on the links and explore the web interface. I recommend starting with your home project's top level 'overview' page - click on the Overview tab and you may see something like this:

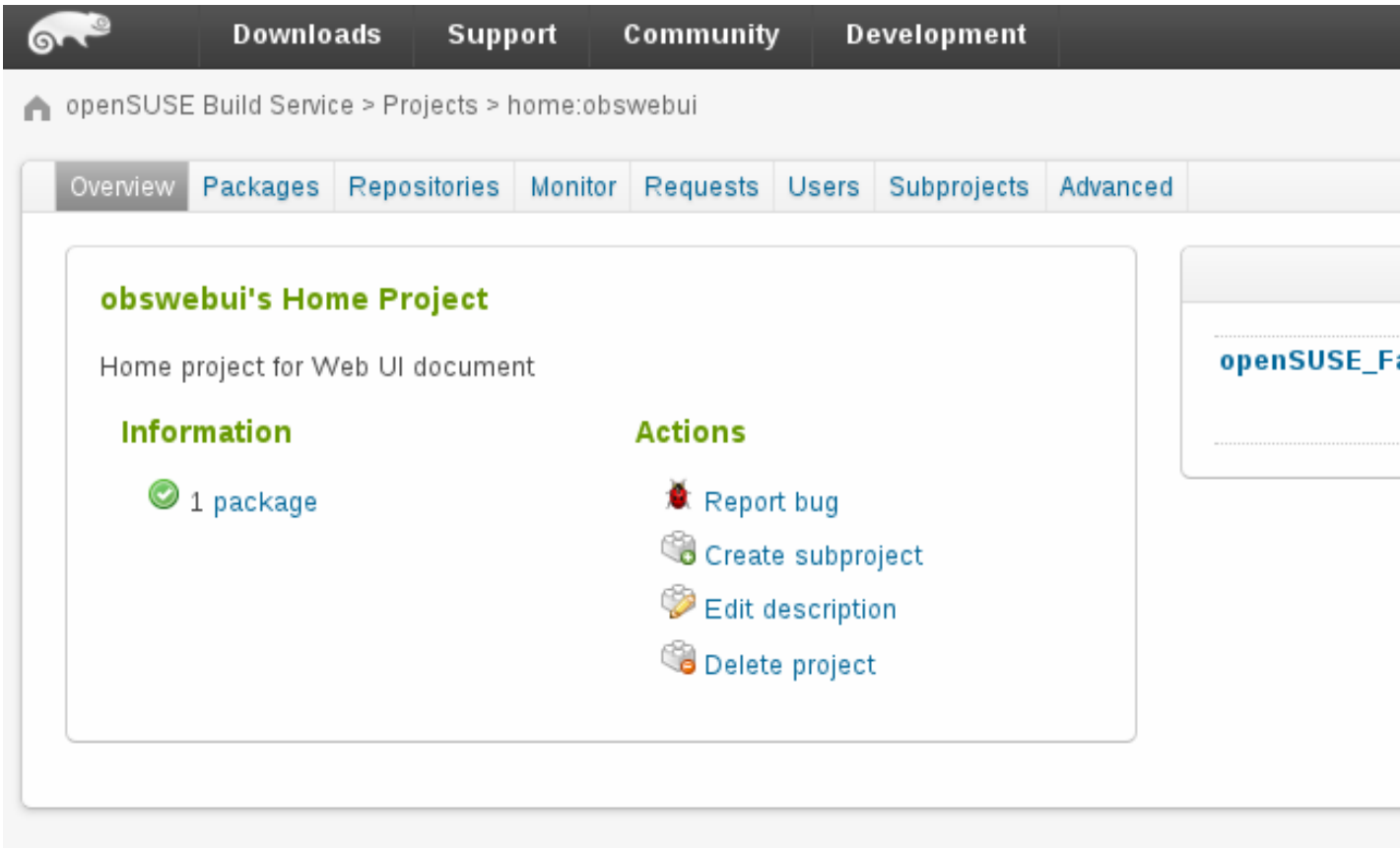

FIGURE 1.13: FLOOD_SUCCEEDED_FINISHED

If you wait a bit, you would see the below building success screen

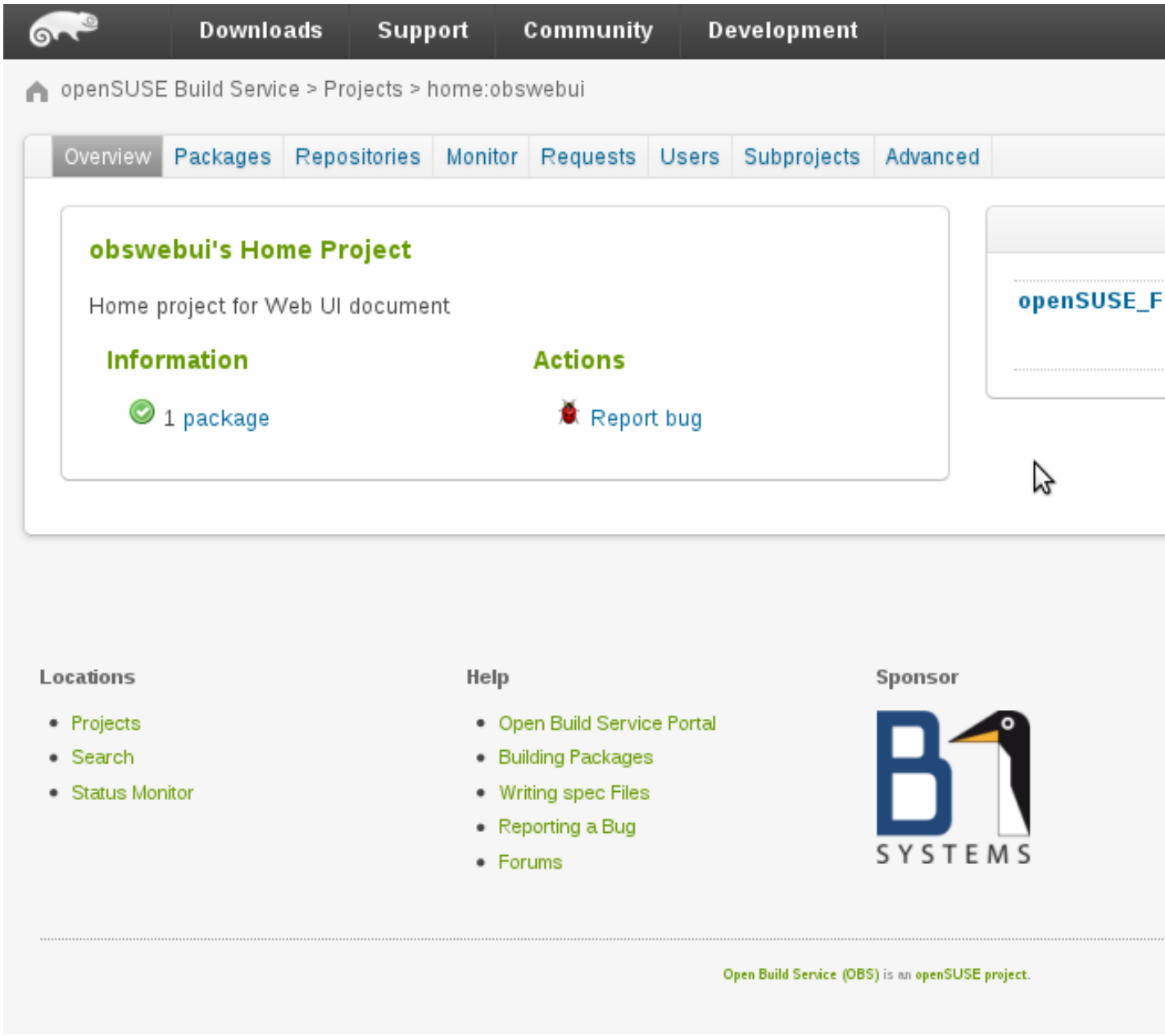

FIGURE 1.14: FLOOD_BUILD_SUCCESS

Click the “succeeded” message, then you will see the build log as below.

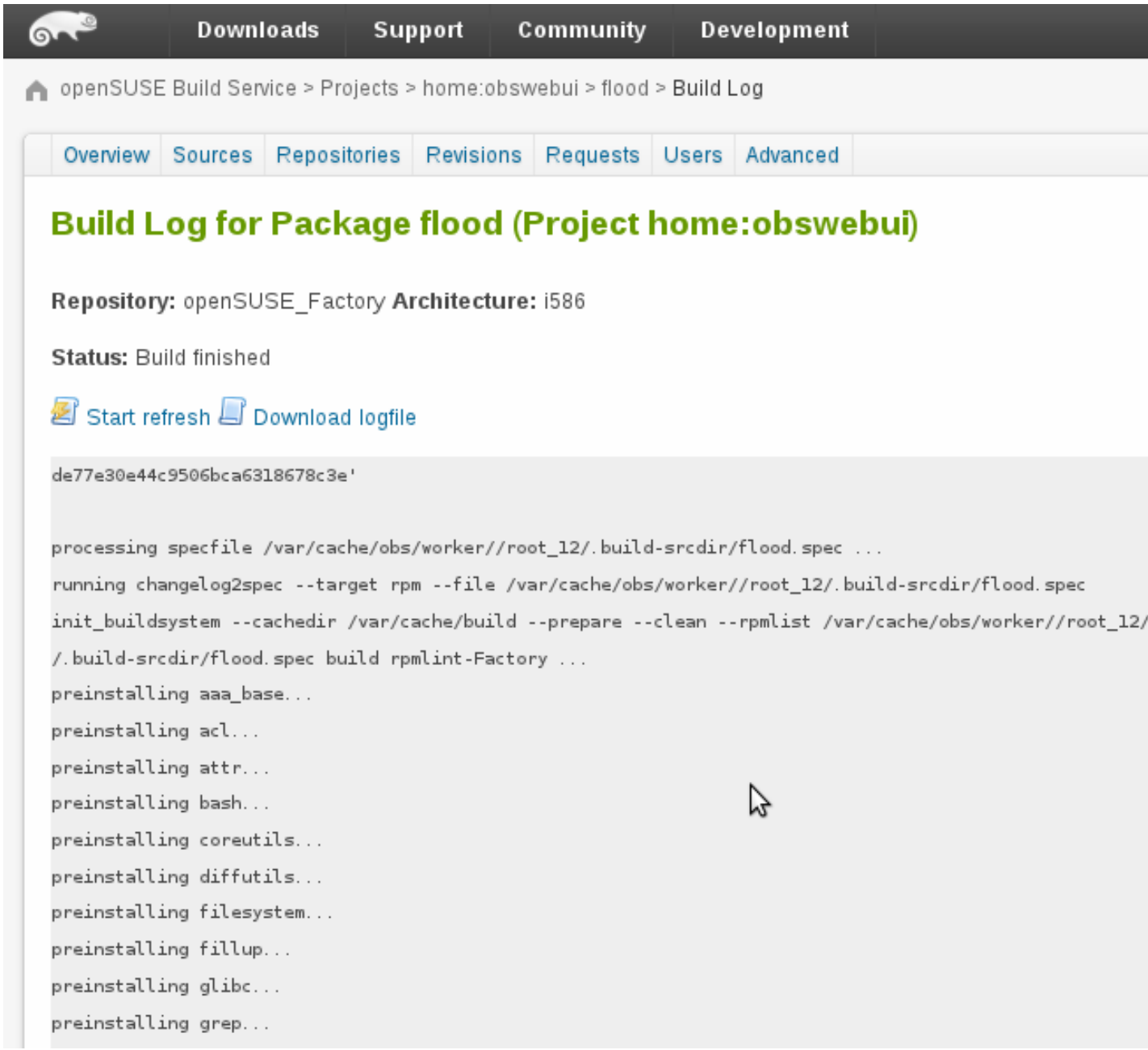

FIGURE 1.15: FLOOD_BUILD_LOG

1.5 Repository Output: Built Packages

To find the RPMs you built, go to your home project page and click Repositories. From there click on the blue repository name. For example, openSUSE_Factory:

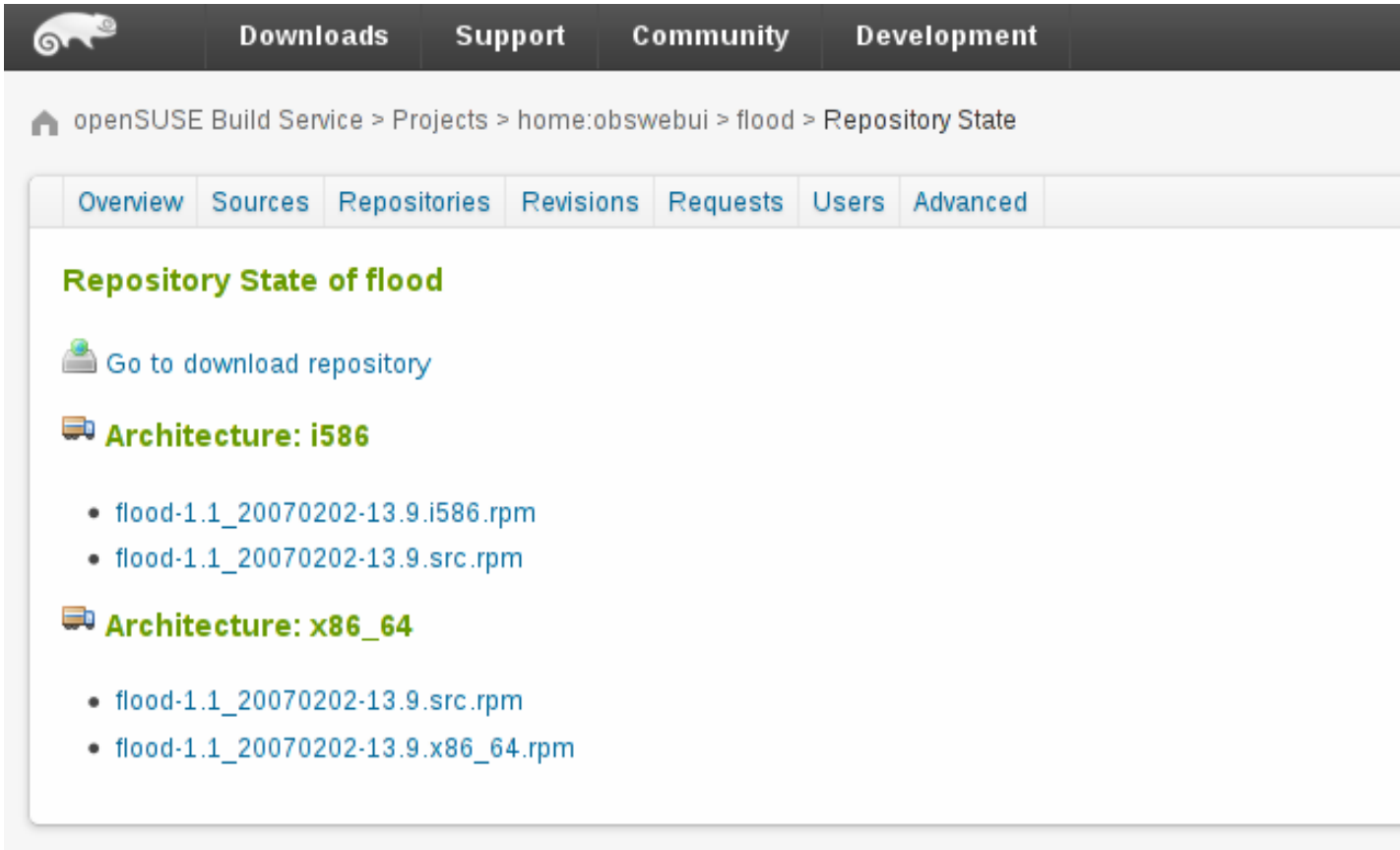

FIGURE 1.16: MY_REPOSITORY

Note

Published repositories are marked with the OBS truck

Now click *Go to download repository*. Note that publishing the repository might take a while. Before the binary repository is published, you will receive a 404 error. When the binaries are available, you will see something like this:

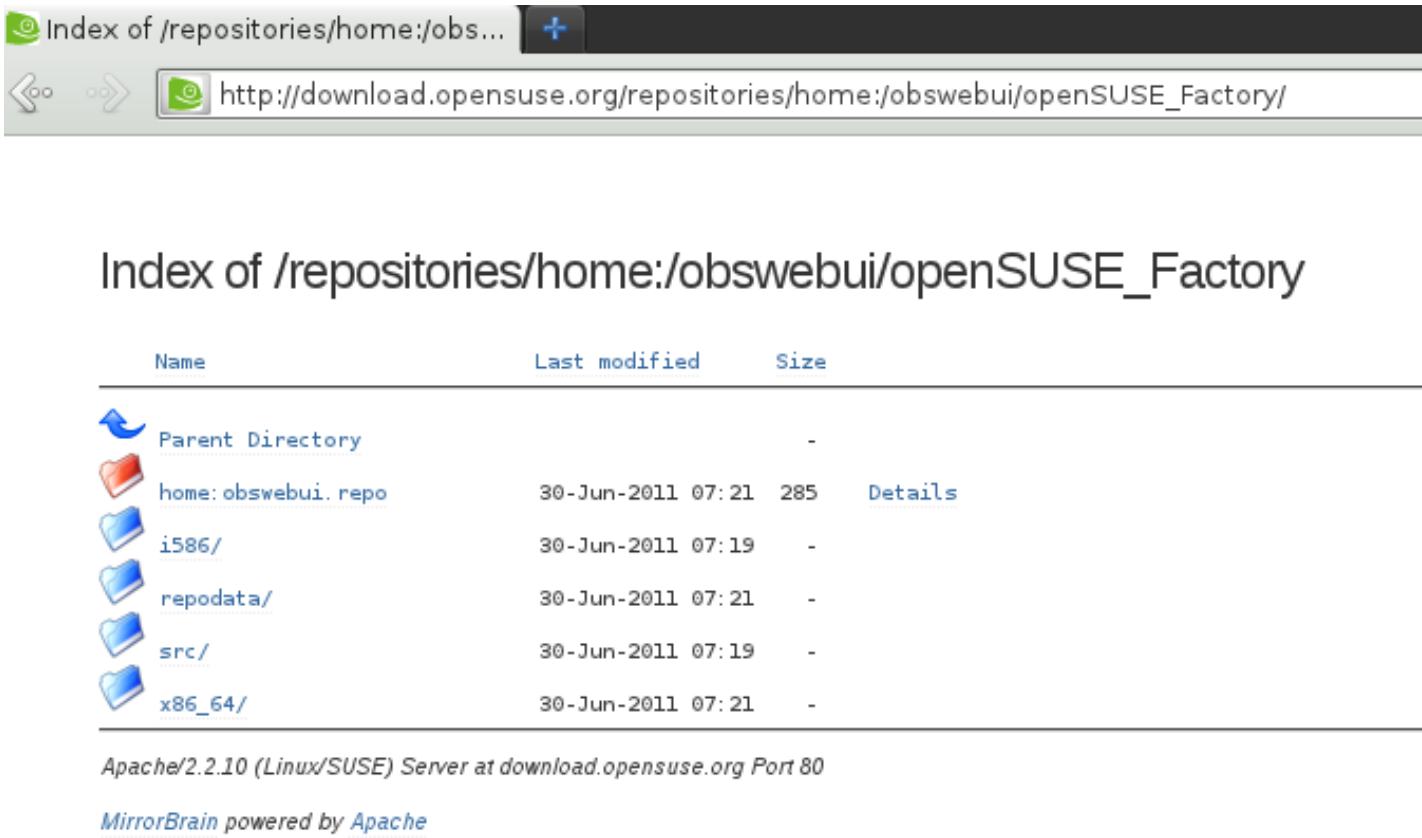

FIGURE 1.17: REPOSITORY STRUCTURE

Your rpms can be found in the subdirectories, and the .repo file is suitable for use with zypper, yum or other repository-friendly package management tools.

1.6 Managing Repositories

This section will show how you can manage your project's repositories.

1.6.1 Adding a repository

To add a repository to your project, click on "Add Repositories" on the project's repository tab. This will direct you to a list of possible distributions you can build packages for, see *Figure 1.18, "Adding a Repository to a Project"*.

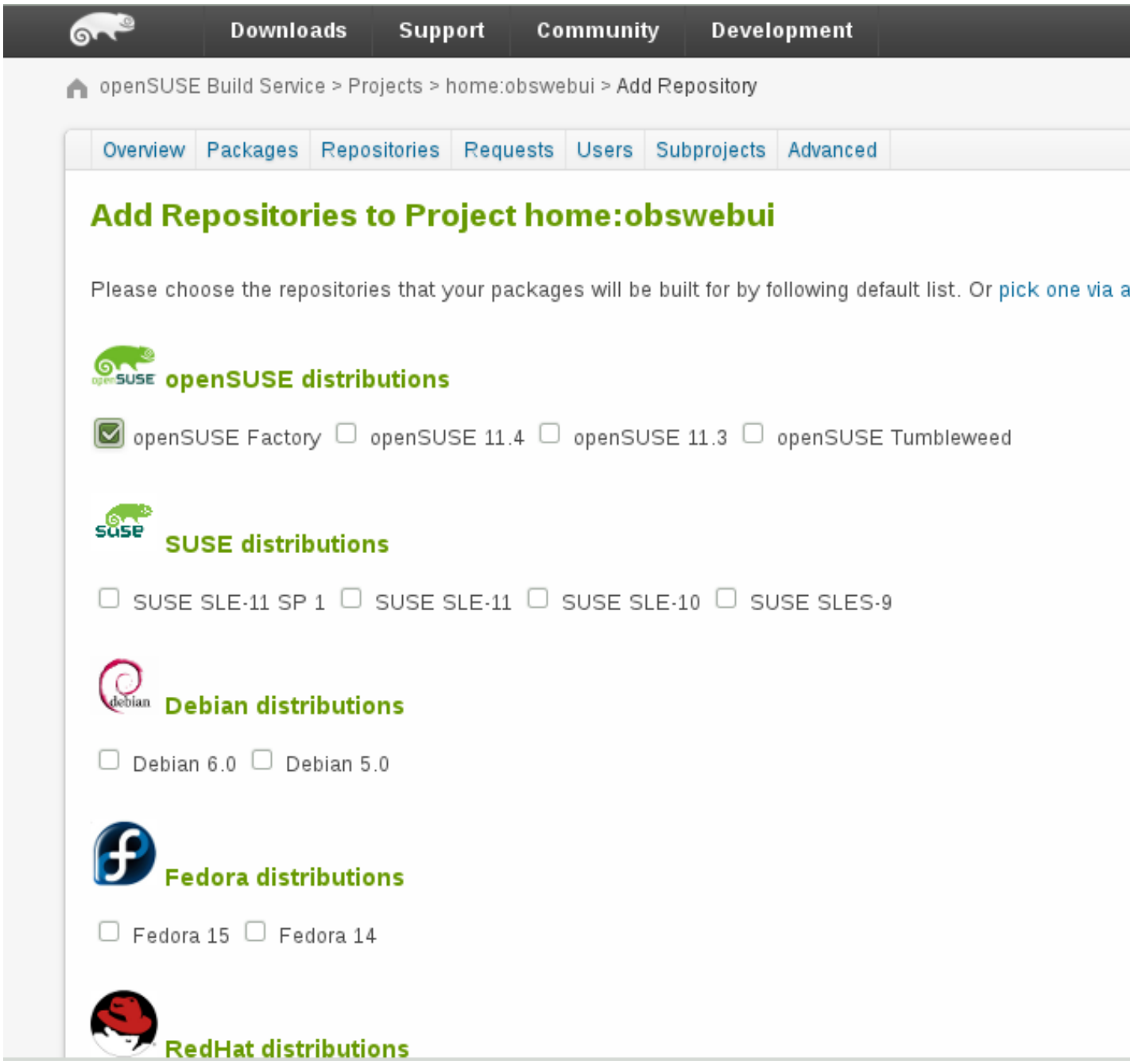

FIGURE 1.18: ADDING A REPOSITORY TO A PROJECT

Note

If you could not find a repository that fits your needs, you might want to switch to the expert mode. Click on the "Expert mode" link right to the button. This page allows you to search and select a repository of any project available in OBS and add it to your projects repository list.

This will take you back to your home: project. The build repository might be disabled: if so, click on the cogwheel to enable it. Congratulations, it is configured. On a heavily loaded server, it can sometimes take a few minutes for your changes to become effective, but your linked package will automatically begin building.

1.6.2 Add Download on Demand repositories to a project

When you have administrator rights you will be able to add Download on Demand repositories to your project. To do so, click on the "Add DoD repository" link and enter your DoD repository data into the form.

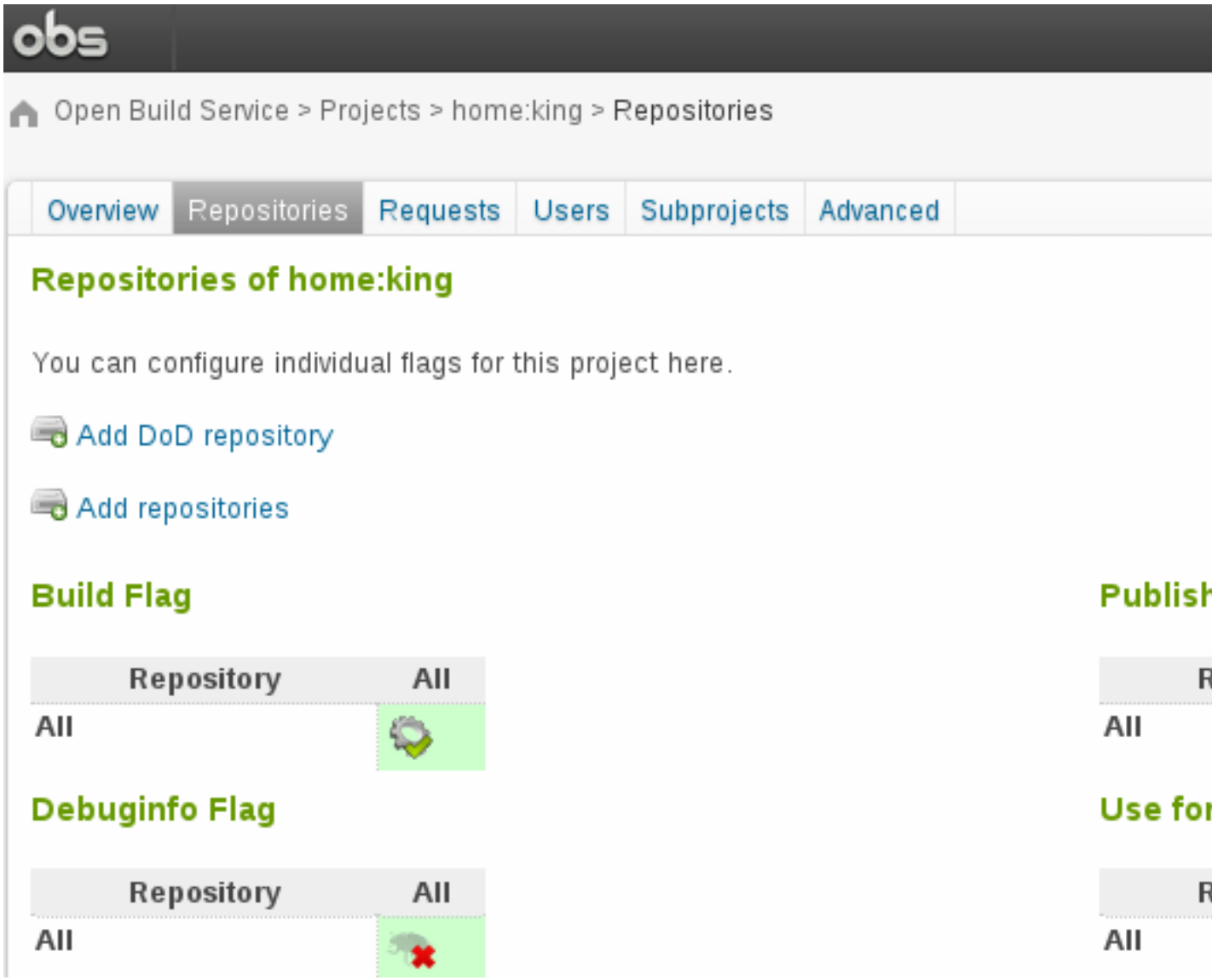

FIGURE 1.19: ADDING A DOWNLOAD ON DEMAND REPOSITORY

The minimal set of fields you have to enter are architecture, repository type and the URL that provides the binary packages. Detailed information about the data you can enter here you can find at the DoD concept section (http://openbuildservice.org/help/manuals/obs-reference-guide/cha.obs.concepts.html#concept_dod). Press "Save" to create the repository.

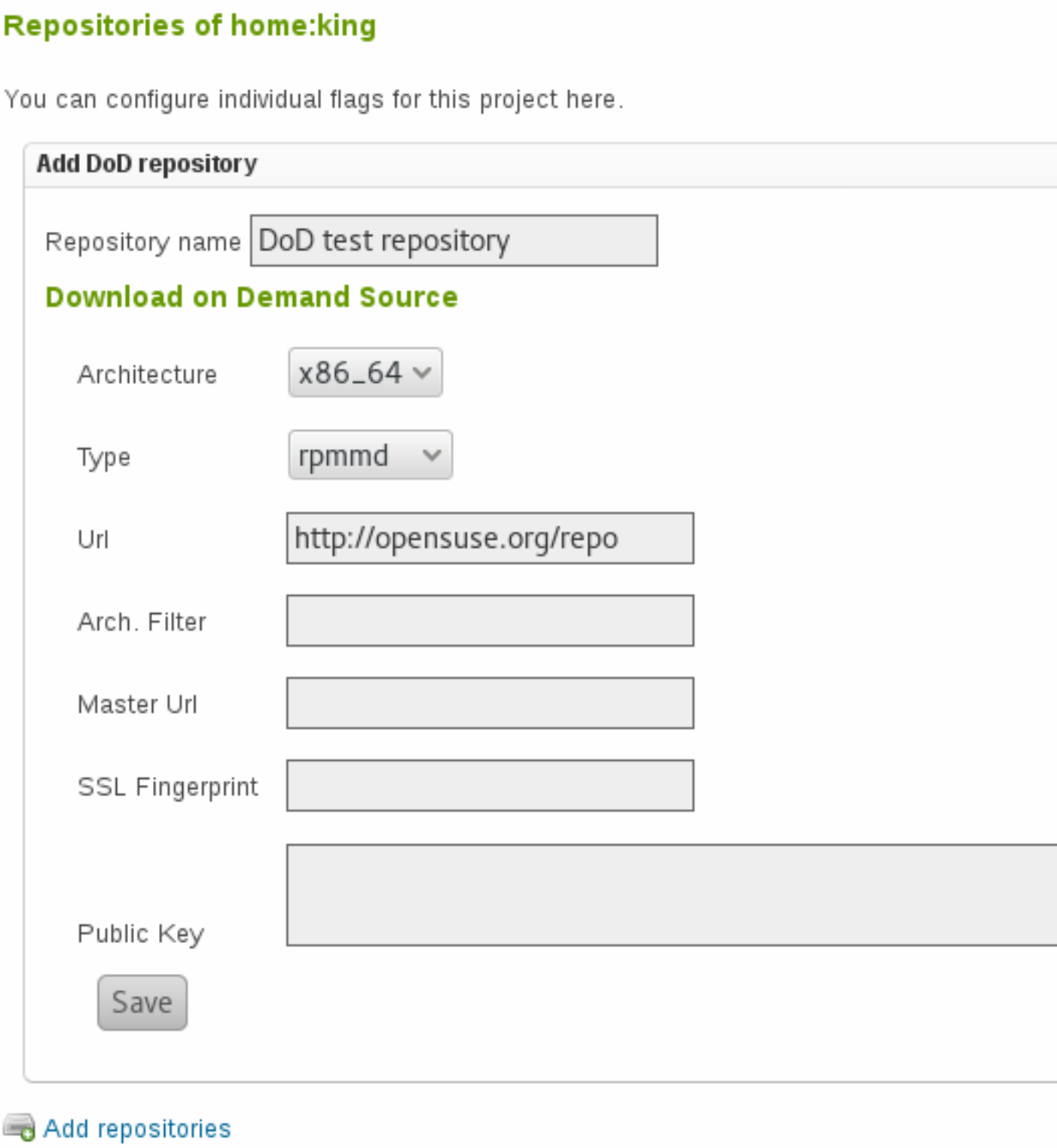

FIGURE 1.20: DOWNLOAD ON DEMAND REPOSITORY FORM

When the repository got added you are able to edit, delete or add additional DoD repository sources.

1.6.3 Adding DoD Repository Sources to a Repository

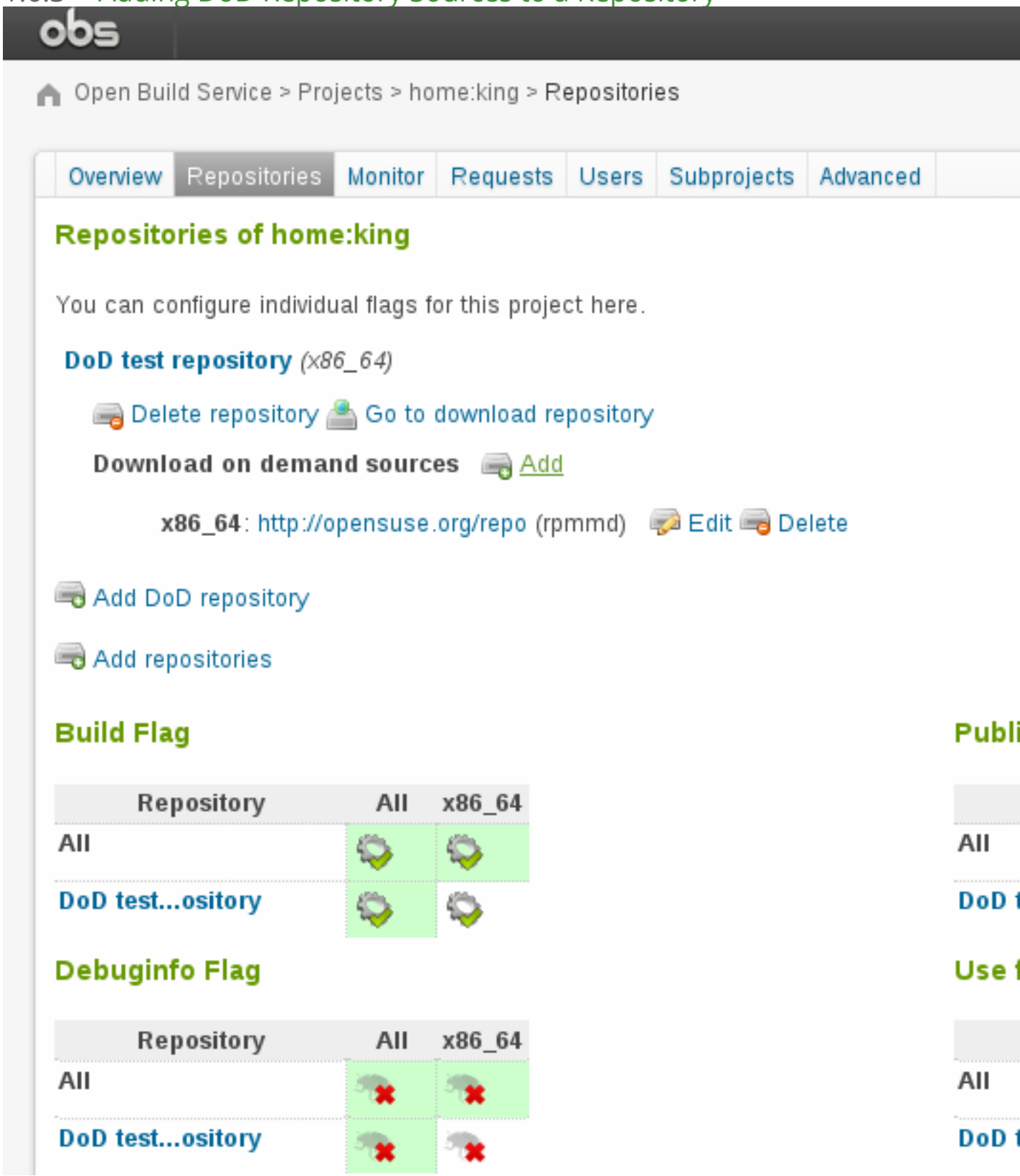

FIGURE 1.21: ADDING DOWNLOAD ON DEMAND REPOSITORY SOURCES

Open the DoD repository sources form by clicking the "Add" link. Here you can enter your additional DoD repository source. Click the "Add Download on Demand" button.

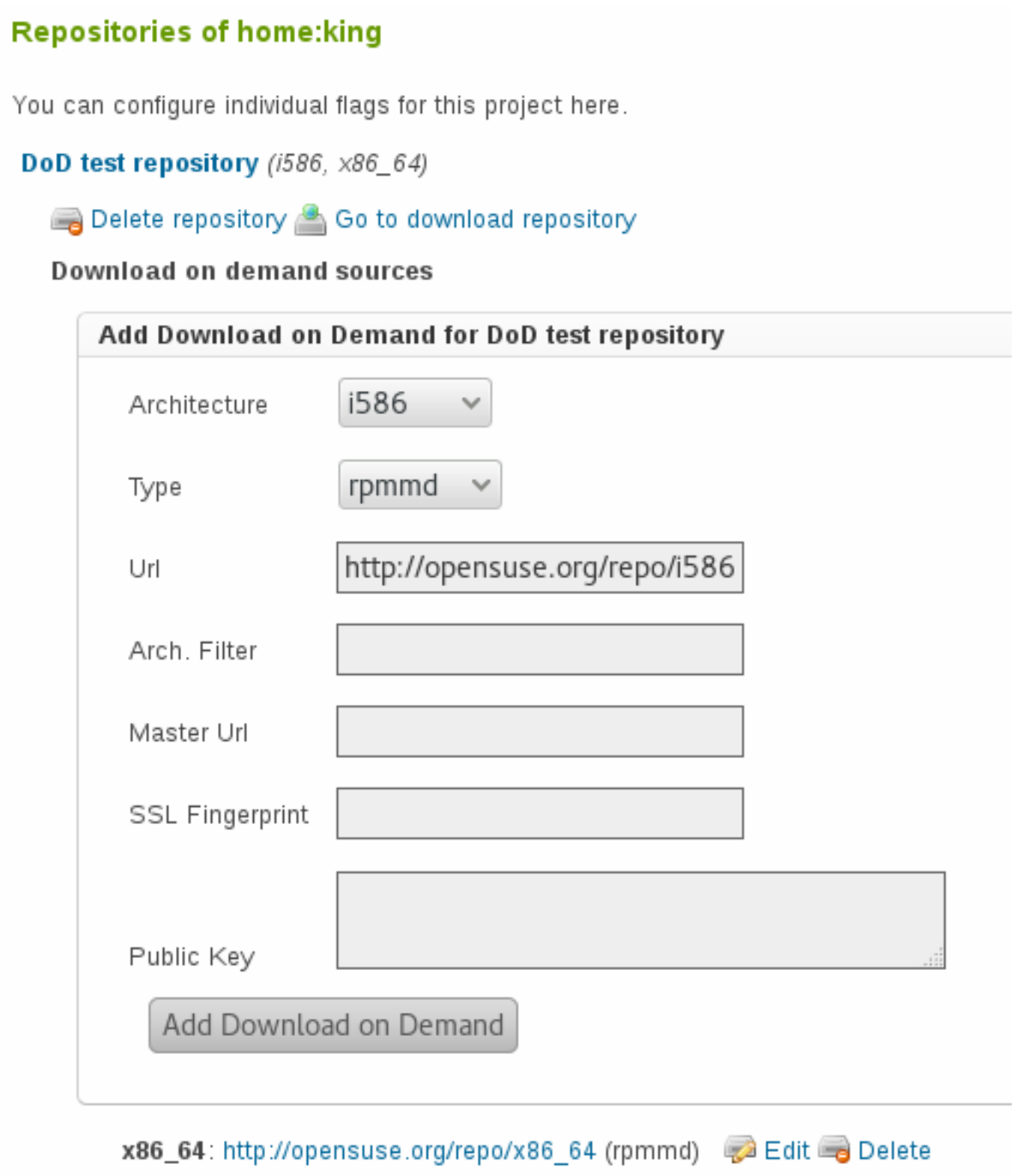

FIGURE 1.22: FORM FOR ADDING DOD REPOSITORY SOURCES

1.6.4 Editing DoD Repository Sources

To edit DoD repository sources after they got added click on the "Edit" link that you find right to each DoD repository source.

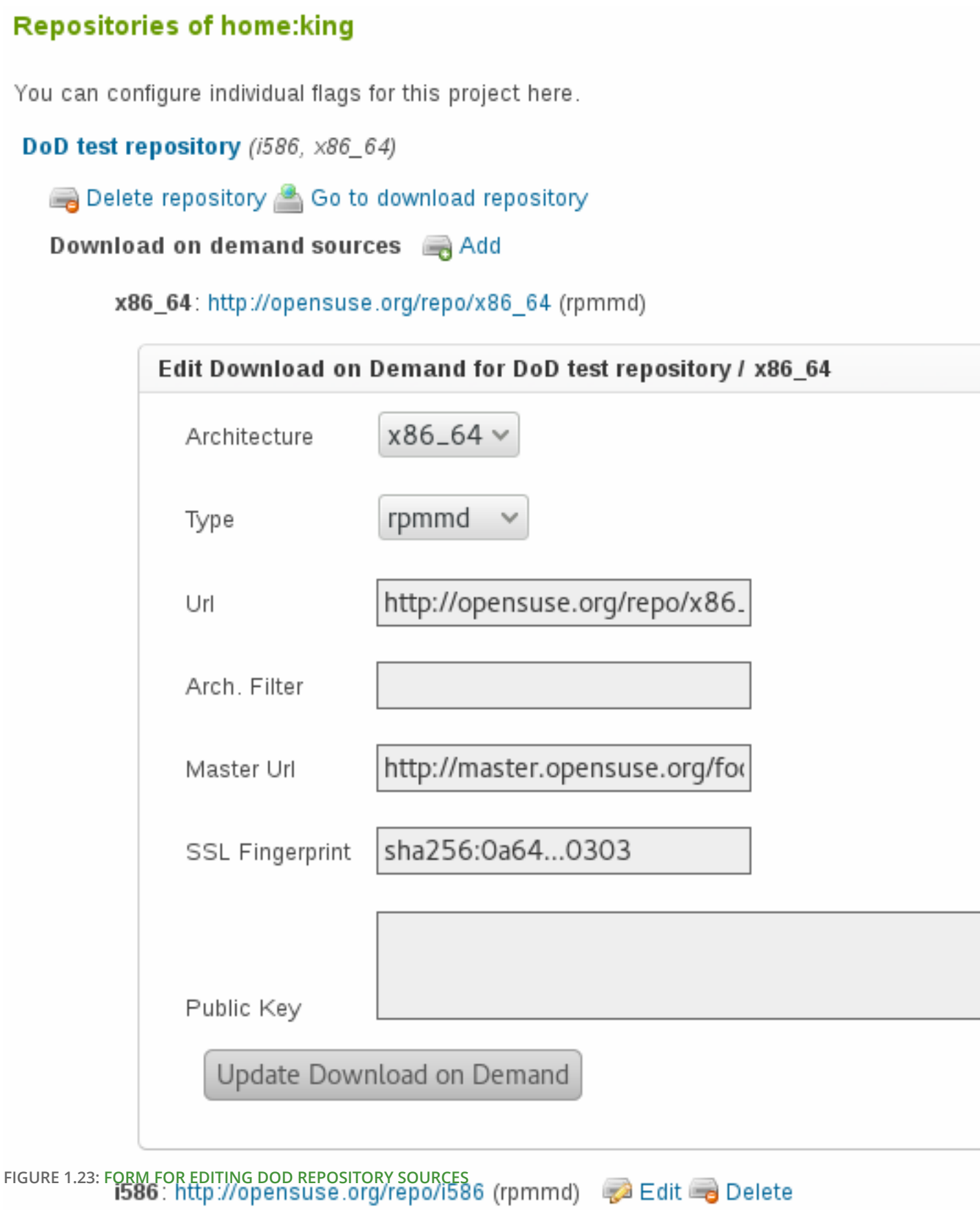

FIGURE 1.23: FORM FOR EDITING DOD REPOSITORY SOURCES

1.6.5 Editing DoD Repository Sources

To delete a repository or repository source, click on the "Delete" link and accept the confirmation dialog.

1.7 Image Templates

Image templates are pre-configured image configurations. The image templates page (https://build.opensuse.org/image_templates) provides a list of these templates. Users can clone these templates and further configure them as they like.

How you can create your own image templates will be shown here.

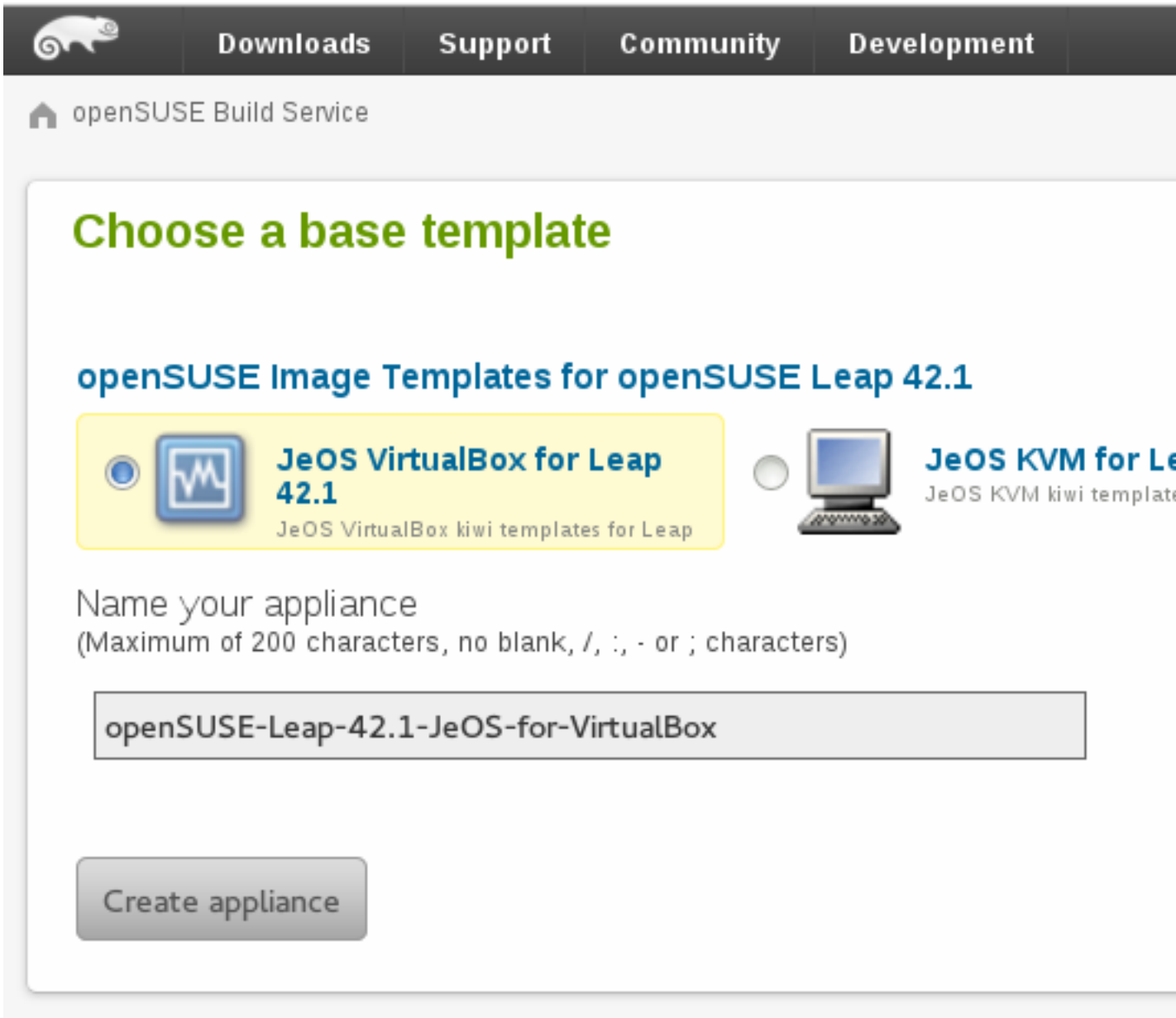

FIGURE 1.24: OBS TEMPLATES PAGE

1.7.1 Creating Own Image Templates

Create a subproject of your home project.

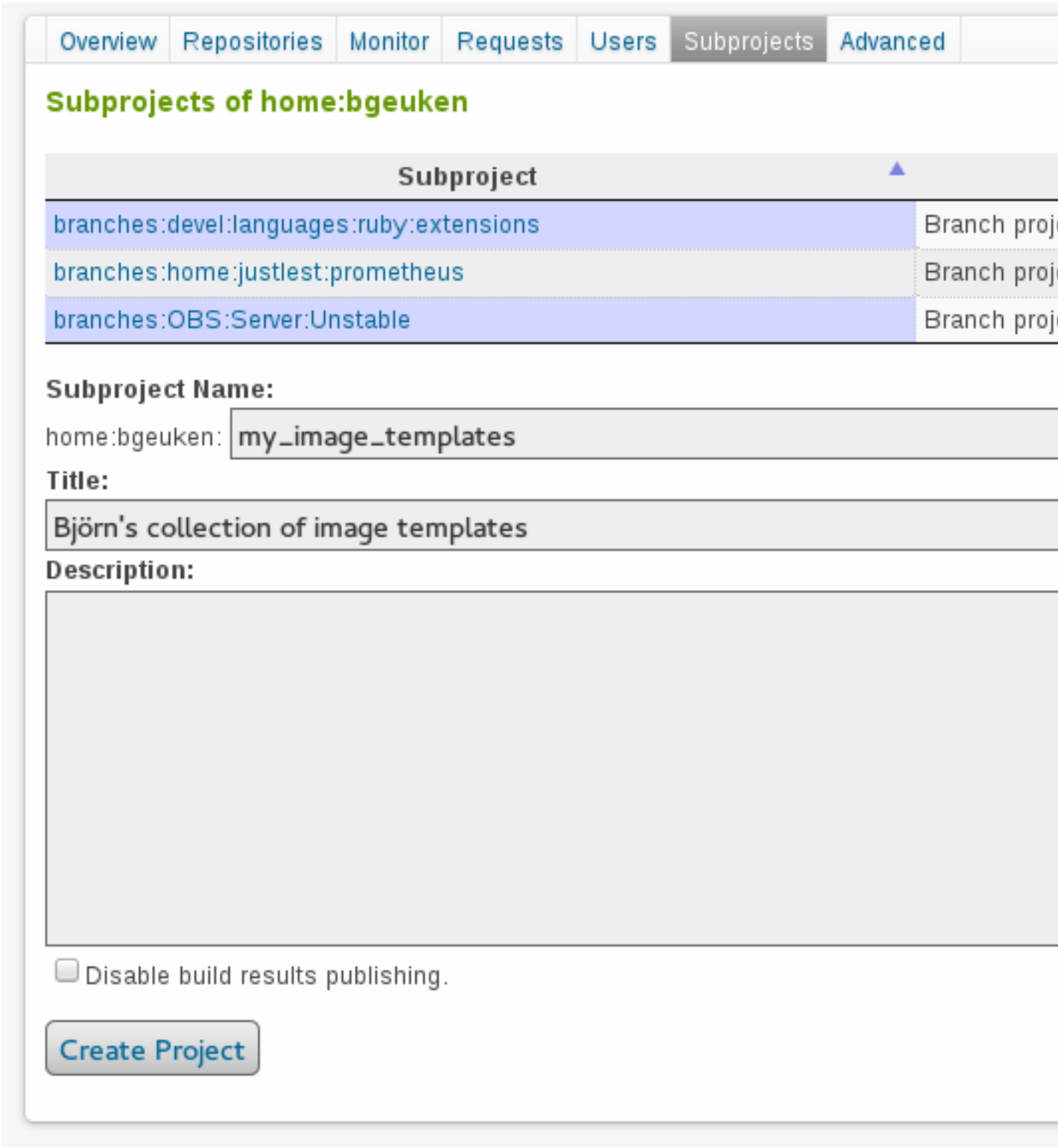

FIGURE 1.25: FORM FOR CREATING IMAGE TEMPLATE SUBPROJECT

Note

Published image templates are fetched via a project's attribute. Any package container living in a published project will be visible on the image templates page.

Within that project create a new package. That will be your actual image template.

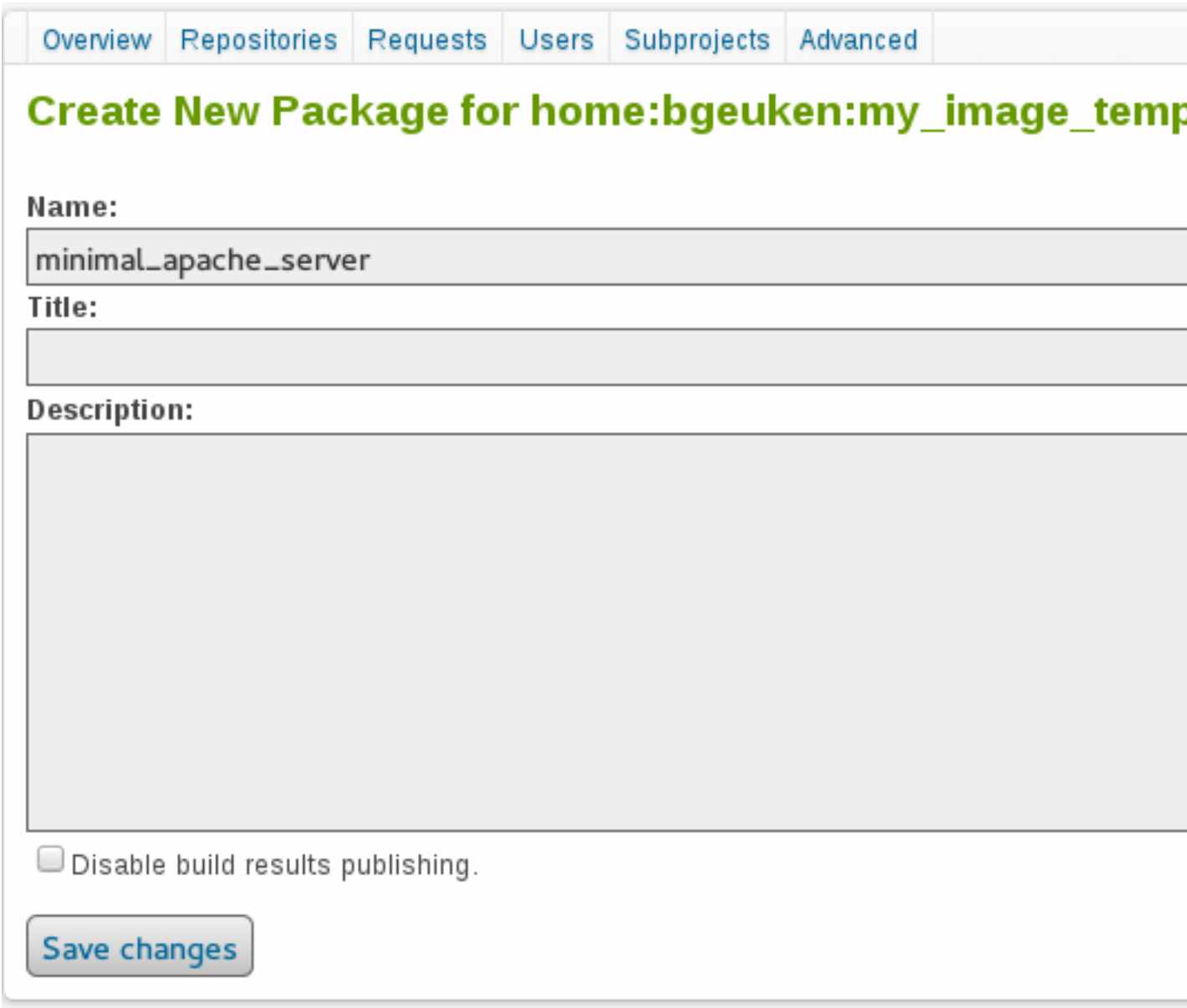

FIGURE 1.26: NEW IMAGE TEMPLATE

Add the 'KIWI image build' repository to your project. This repository is needed to build KIWI images in your project. Go to the 'Repositories' tab, click on 'Add repositories' and click on the 'KIWI image build' check box.

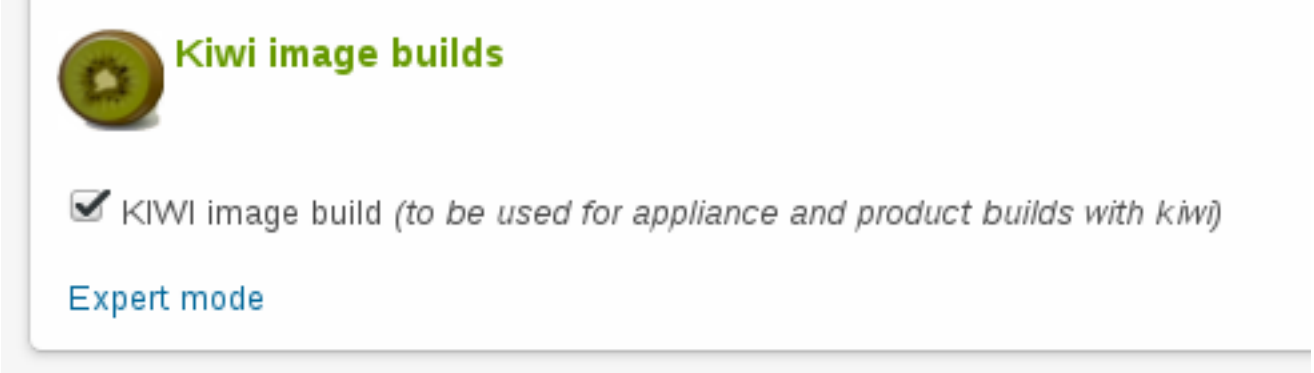

FIGURE 1.27: ENABLING THE KIWI IMAGE BUILD REPOSITORY

Add sources for your image configuration.

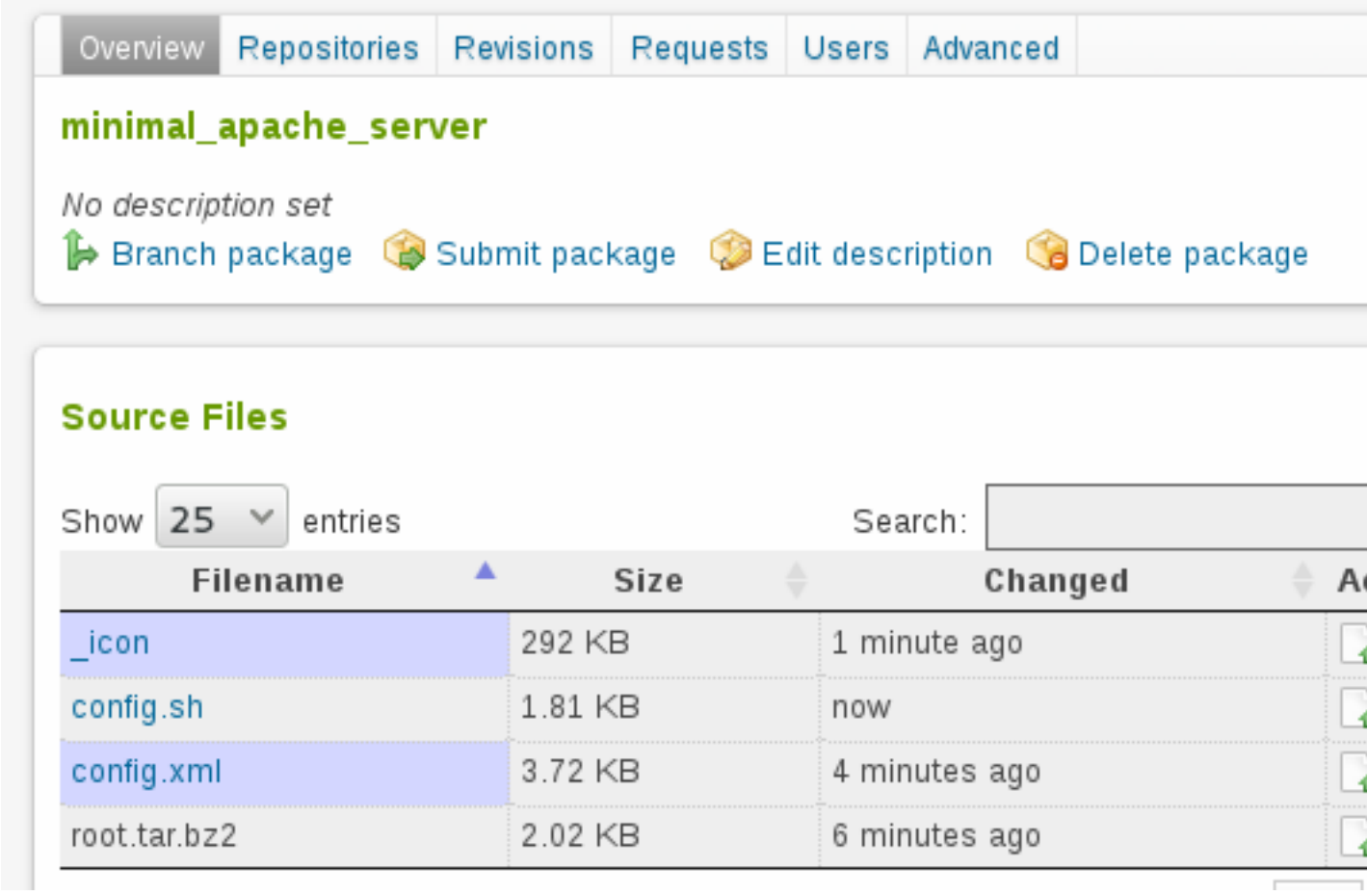

FIGURE 1.28: OVERVIEW OF SOURCES OF A CUSTOM IMAGE TEMPLATE

KIWI configurations usually consists of an xml configuration root tarball.

In addition, you can define an icon for your image templates by adding graphical image (for example, PNG, JPG) to your template sources and name it `_icon`. If that file exists, it will be used as icon for your image on the image templates page.

For a full list of image descriptions and general information about building images with KIWI, see the KIWI project page (http://opensuse.github.io/kiwi/) and the KIWI cookbook (https://doc.opensuse.org/projects/kiwi/doc/).

1.7.2 Publishing Image Templates on the Official Image Templates Page

Once everything is set up and your templates are building, you might want to publish them. In that case contact the admin of the OBS instance you are using and ask them kindly to do so. If you happen to use the official OBS (https://build.opensuse.org/), that would be admin@opensuse.org.

1.8 KIWI Editor

You can edit the KIWI file associated to your project. It is only possible, at the moment, to edit the repository list and packages with type image. If you are running your own instance of OBS be sure you have the kiwi_image_editor feature enabled in your config/feature.yml file.

1.8.1 Accessing the KIWI Editor

Go to your package, and upload a file with the `.kiwi` extension (for example, `test.kiwi`), with valid KIWI content.

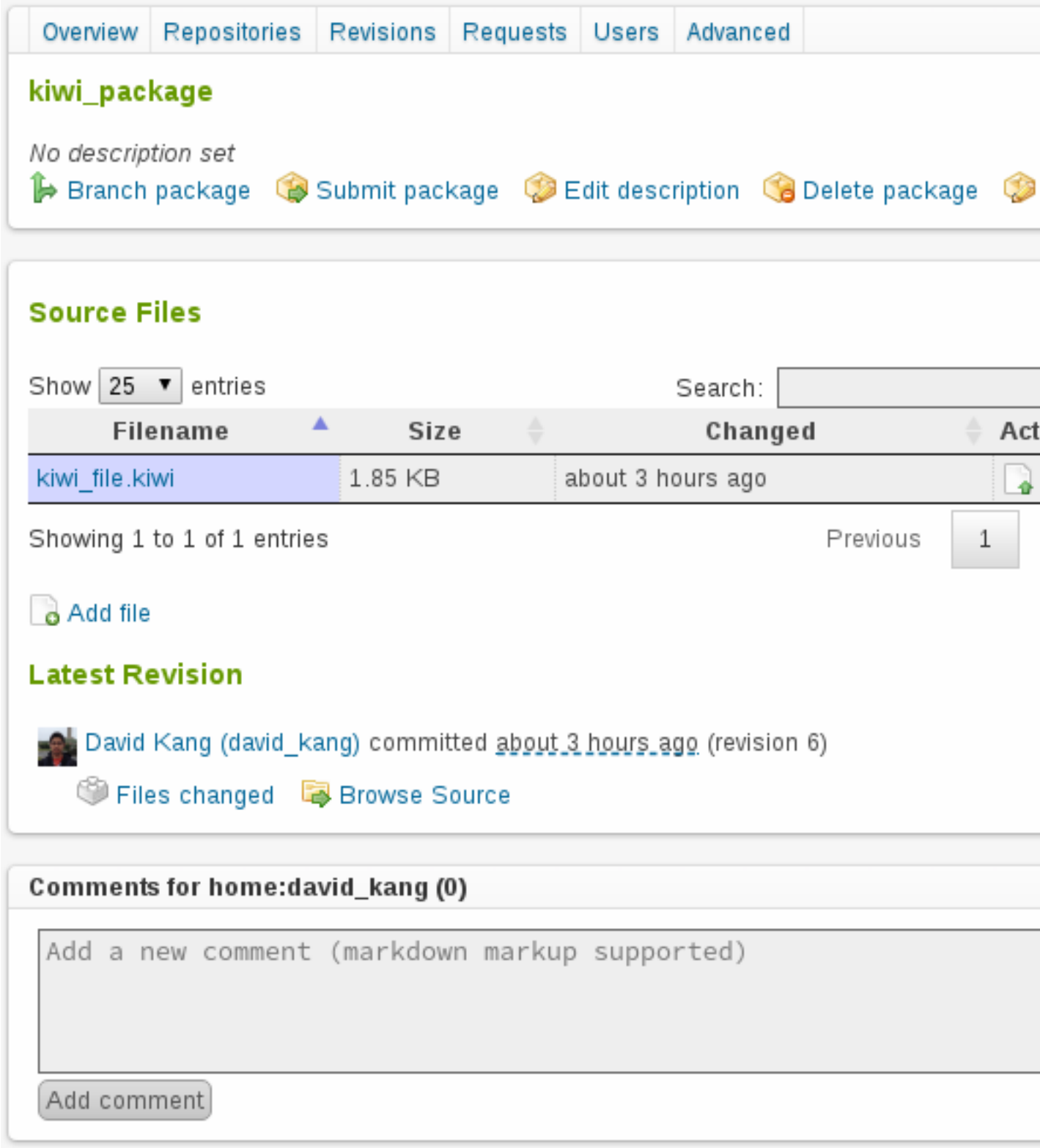

FIGURE 1.29: EXAMPLE OF A PACKAGE WITH A KIWI XML FILE

Note

You should see now a "Edit KIWI" link (next to "Delete package" link).

Click on the "Edit KIWI" link and you will be redirected to the Editor.

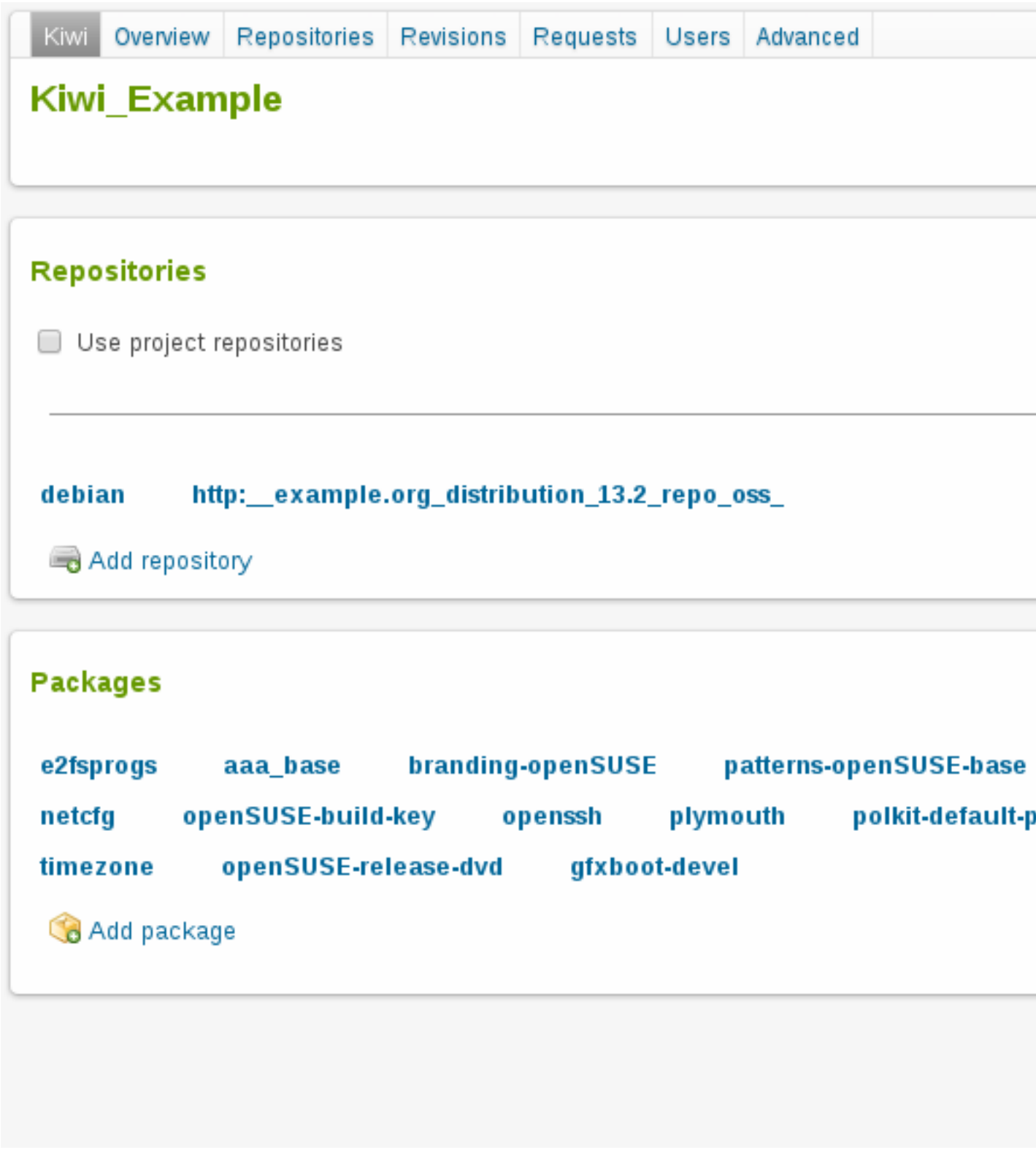

FIGURE 1.30: KIWI EDITOR. SHOW SCREEN

- *Repositories:* Displays the repositories set in the Kiwi file.
- *Packages:* Displays the packages of the package group with type *image*.

1.8.2 Adding Repositories in the KIWI Editor

To add a new repository click *Add repository* link and fill in the dialog. There are two ways to create it:

- *Basic Mode:* Adding the name of a project will provide a list with the repositories from that project.

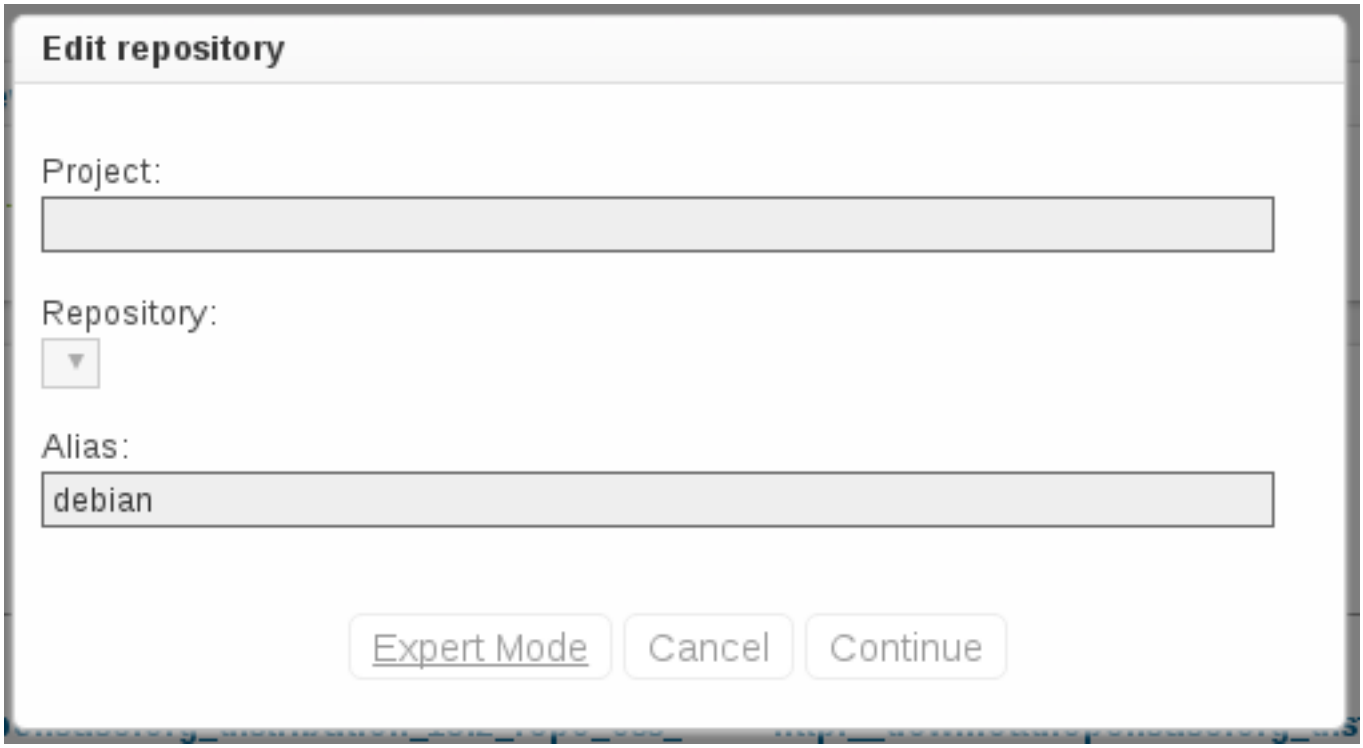

FIGURE 1.31: KIWI ADDING A NEW REPOSITORY - BASIC MODE

- *Expert Mode:* This mode provides you with a set of customizable parameters for creating a repository.
  - *Type:* Valid options are *rpm-md* and *apt-deb*.
  - *Priority:* Repository priority for the given repository.
  - *Alias:* Alternative name for the configured repository.

- *Source Path:* Define the repository path.
- *User:* Specifies a user name for the given repository.
- *Password:* Specifies a password for the given repository.
- *Prefer License:* The repository providing this attribute will be used primarily to install the license tarball if found on that repository.
- *Image Include:* Specifies whether the given repository should be configured as a repository in the image.
- *Replaceable:* Defines a repository name which may be replaced by the repositories specified in the image description. This attribute should only be applied in the context of a boot image description.

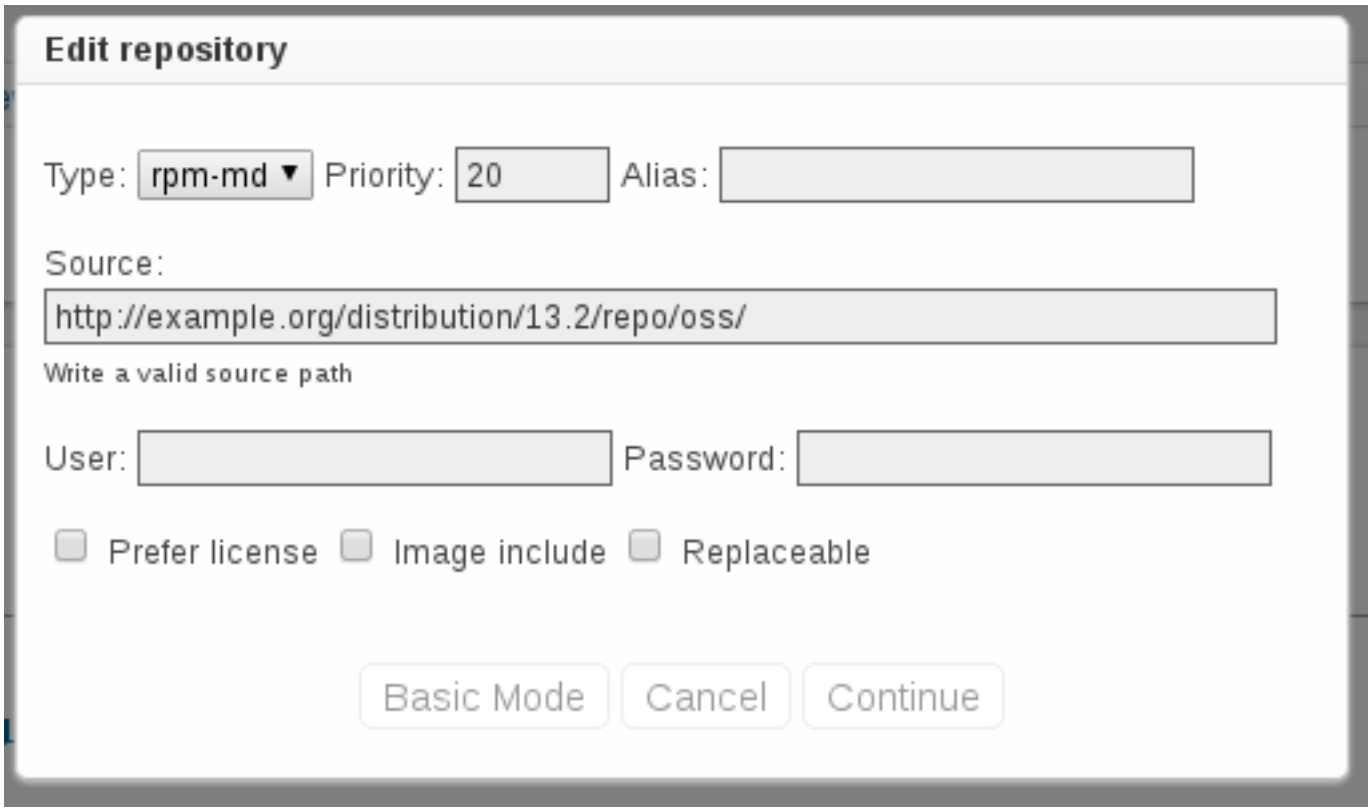

FIGURE 1.32: KIWI ADDING A NEW REPOSITORY - EXPERT MODE

To use the configuration of the current project check the *Use project repositories* checkbox.

FIGURE 1.33: KIWI USE PROJECT CONFIGURATION

Note

This option will remove the other repositories from your kiwi file.

1.8.3 Adding Packages in the KIWI Editor

Adding a package is practically the same as adding a repository. We offer an autocomplete for the package name that will show you the package available in the repositories added previously.

FIGURE 1.34: KIWI ADDING A NEW PACKAGE

Note

The package groups shown in the editor are only those with type *image* and the packages will be added in this kind of package group. If it did exist previously the KIWI Editor creates a package group with type *image* for you.

1.9 Staging

1.9.1 Creating a Staging

At the moment, the Staging page is only accessible by hand. "*/staging_workflows/new?project=openSUSE:Factory*" The creation of a Staging will automatically create two staging projects as a subproject of the main project. Before create we need to select a group of managers, they will be in charge of assigning requests to the staging projects and also excluding requests from the Staging.

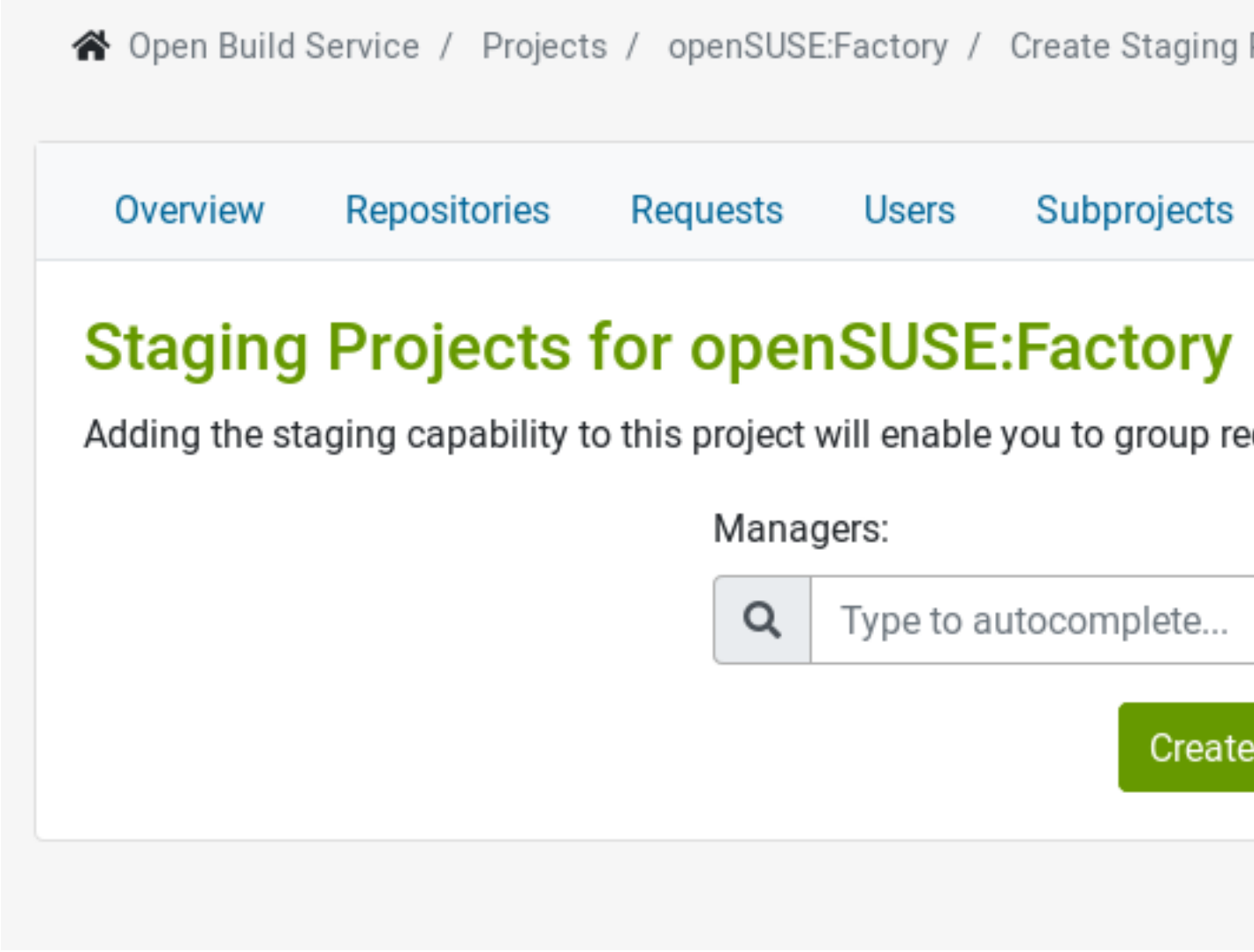

FIGURE 1.35: CREATING A STAGING FOR OPENSUSE:FACTORY

Note

An Admin should previously create the manager groups.

1.9.2 Start using it

In this view, we can find all the staging projects with an associated request and their current state.

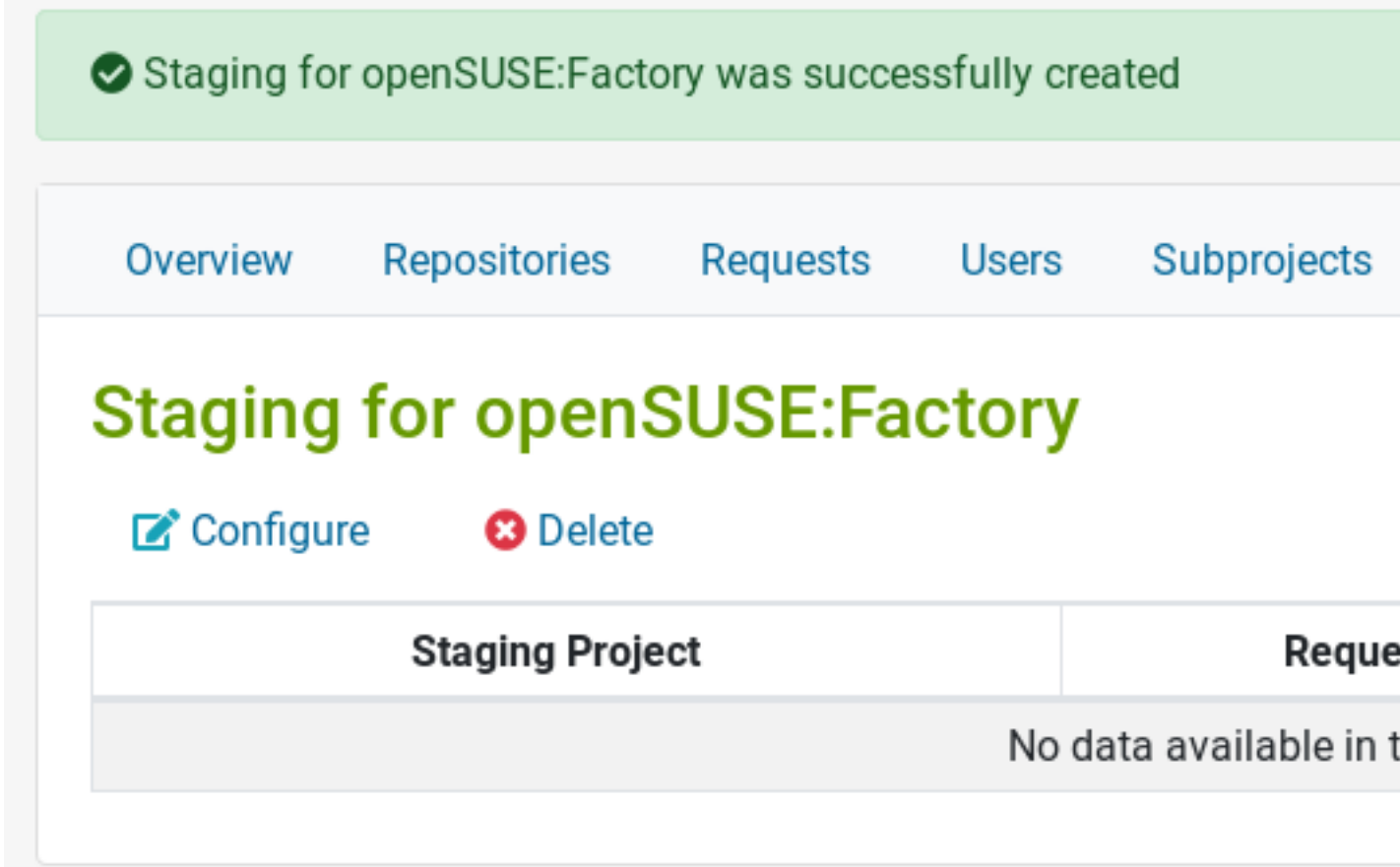

FIGURE 1.36: STAGING SHOW SCREEN

- Table content:
  - *Staging Project:* Shows the staging project name, its overall state (see legend), and the overall build progress of the packages within the project.
  - *Requests:* Show the associated requests and their current state.
  - *Problems:* Shows build problems of packages within the project and status problems reported to the Build Service's Status API by external services like openQA.
- Info section:
  - *Managers:* Shows the Staging Managers group.
  - *Empty projects:* Staging projects without assigned requests.
  - *Backlog:* List of requests that can be assigned to a staging project.
  - *Ready:* List of requests that were in the backlog and have an accepted review.
  - *Ignored:* List of requests excluded from this staging workflow.

1.9.3 Delete a Staging

By clicking on the delete button on the Staging index page, we are able to delete a Staging. By selecting the associated staging projects in the appearing modal window, we are able to delete them as well. If not selected, they will remain as regular subprojects.

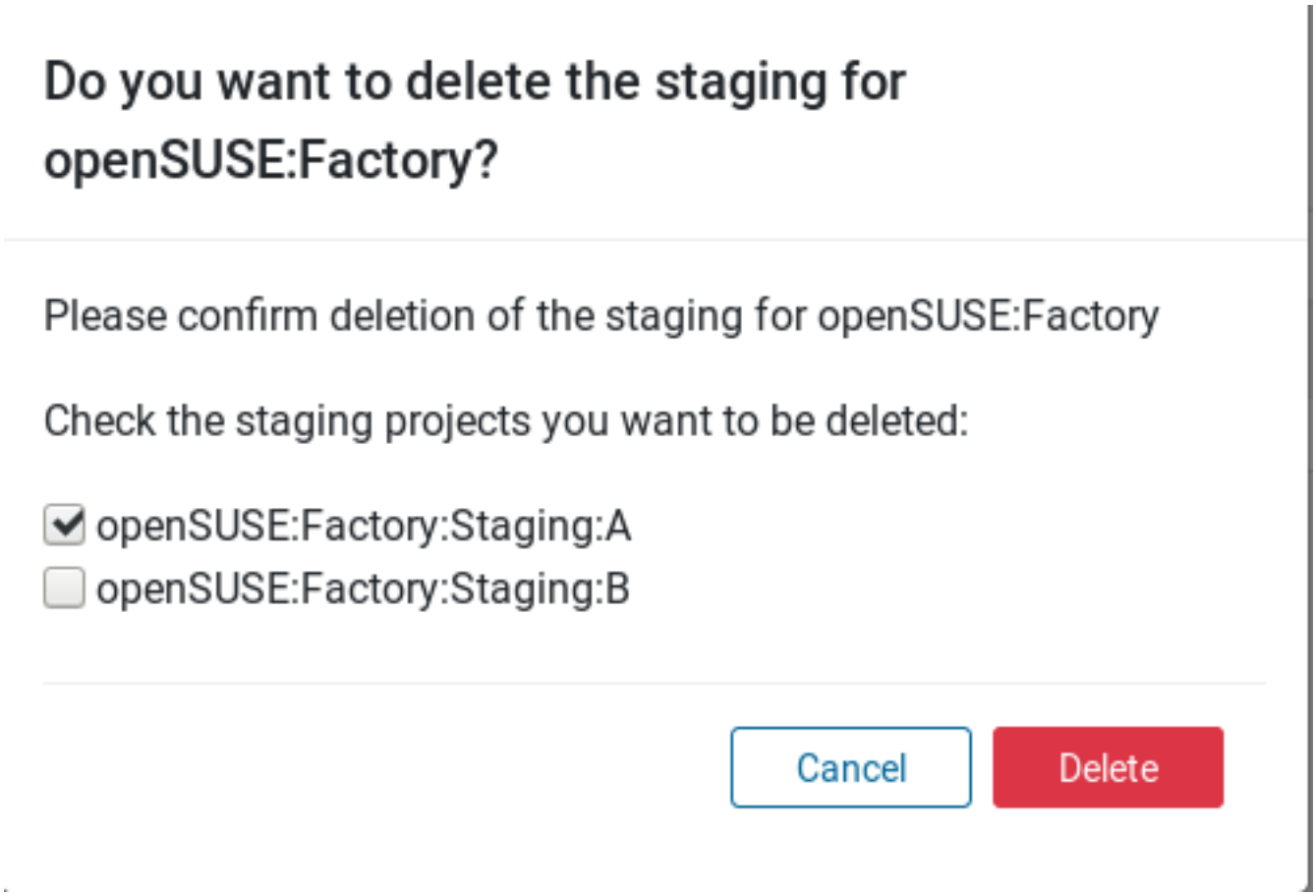

FIGURE 1.37: DELETE A STAGING WORKFLOW

1.9.4 Configuration

It is possible to create/delete staging projects (openSUSE:Factory:Staging:A, openSUSE:Factory:Staging:B, etc) and change the Managers Group of the Staging.

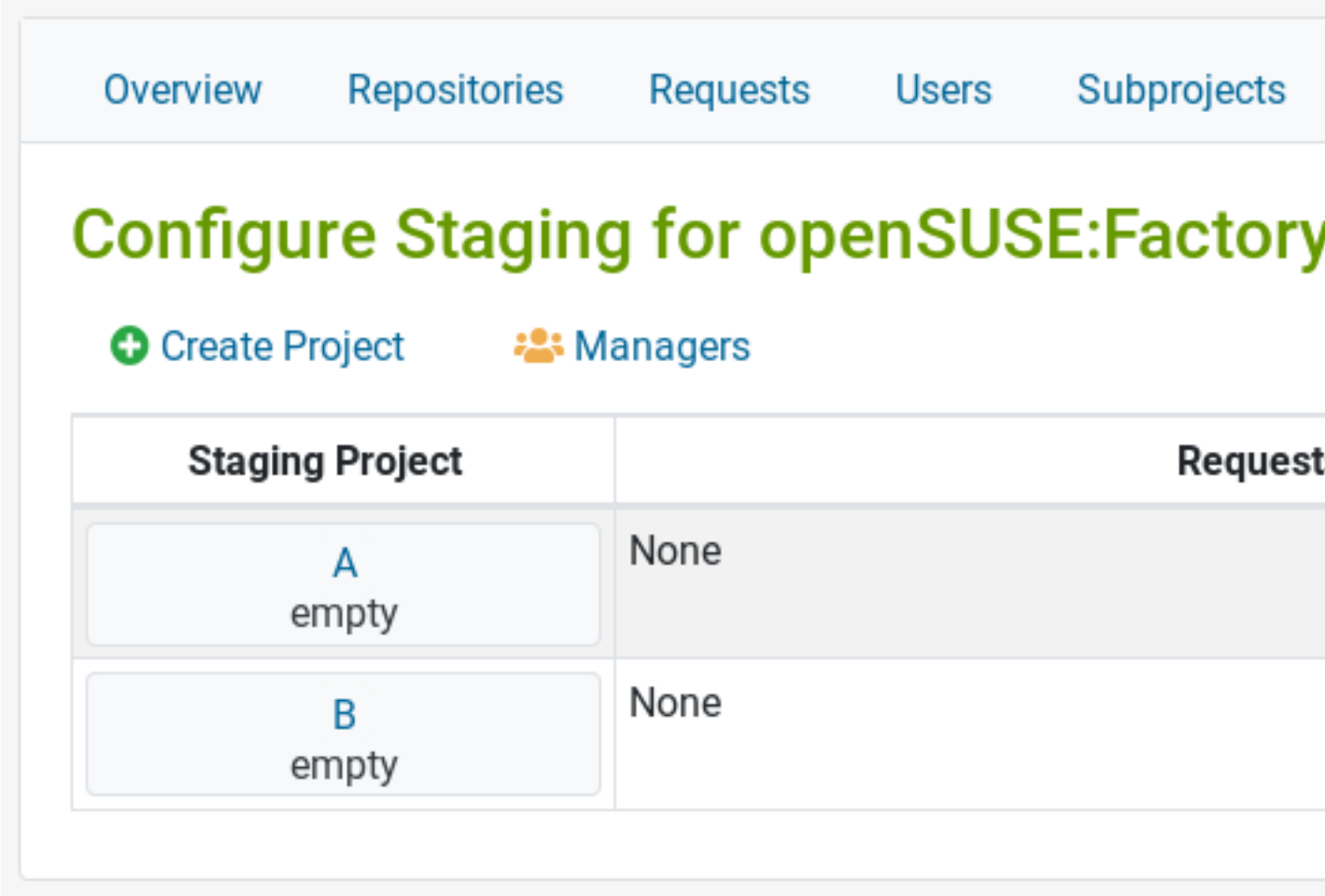

FIGURE 1.38: CONFIGURING A STAGING

Note

Changing the Managers Group of a Stanging will automatically unassign the old group and assign the new group to the related staging projects.

1.9.5 Staging Project

A staging project contains requests assigned by a Staging Manager. An overview page is provided for a staging project where you can find more detailed information about the requests, reviews and checks.

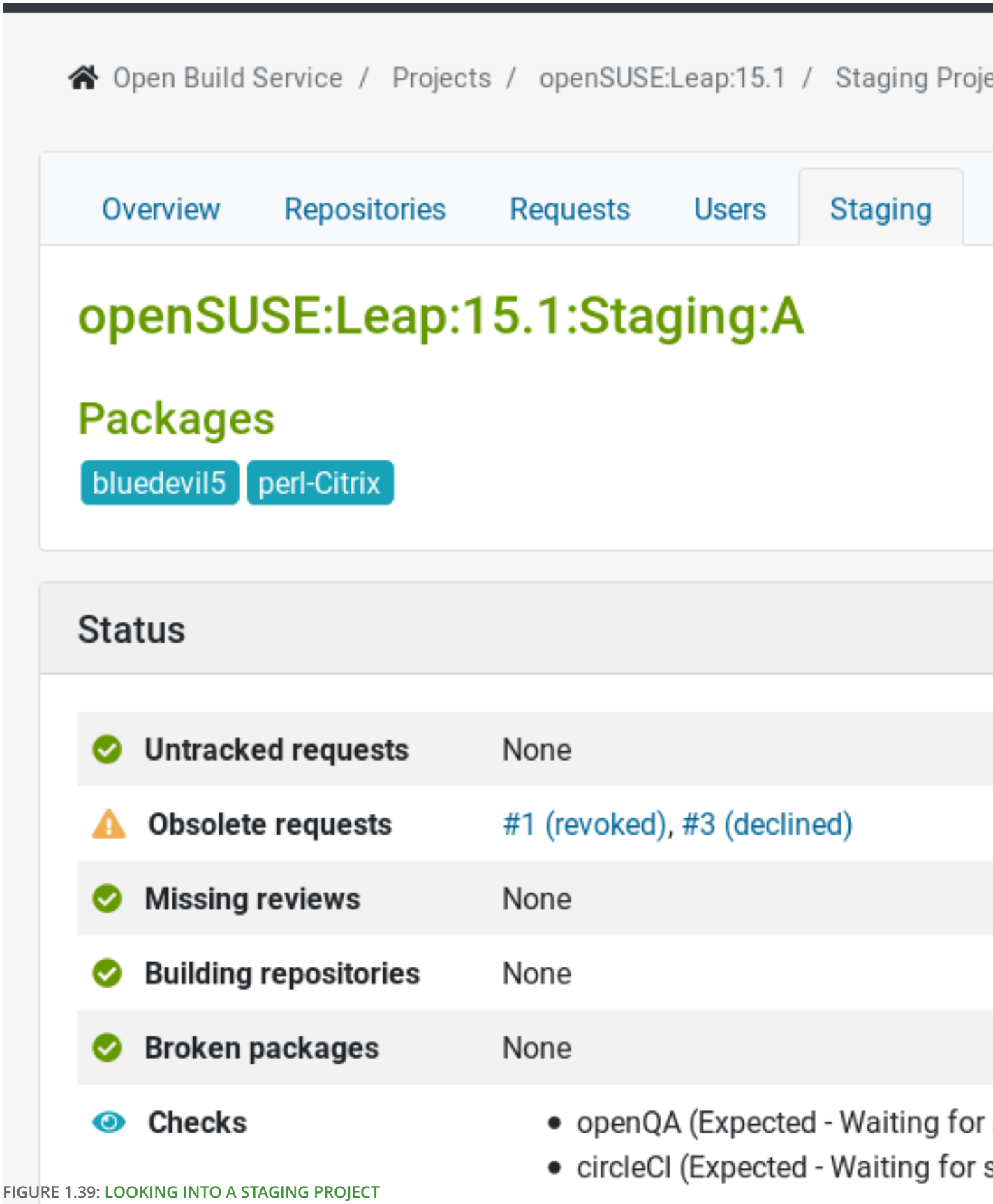

FIGURE 1.39: LOOKING INTO A STAGING PROJECT

- *Obsolete Requests:* Requests that were declined, revoked or superseded.
- *Missing Reviews:* Requests with pendings reviews.
- *Building Repositories:* List of packages that are still building.
- *Broken Packages:* List of packages with failling builds.
- *Checks:* List of checks of the staging project.

2 Basic Concepts and Work Styles

These best practices should be known by every OBS user. They describe how to set up projects and working with own or foreign sources.

2.1 Setup a project reusing other projects sources

You can also setup your own project using the sources, spec files and patches from another project and develop within this project.

```
#osc copypac SOURCEPRJ SOURCEPAC DESTPRJ
```

By default, Open Build Service will strip the maintainer info and now make it part of your own project. To clarify, when we speak of a project, it can mean just one package or a complete set of packages with their own build dependencies.

2.2 Contributing to External Projects Directly

In case a user does not have commit permissions for a project, they can request maintainership permissions for this project. This makes sense if the user is already known to the project owners and they trust them as a maintainer. There is a way to do this via the request system of OBS, but only via osc so far:

```
# osc createrequest -a add_me maintainer PROJECT
```

2.3 Contributing to Foreign Projects Indirectly

Users who are new to a given project, either because they are new users with Open Build Service or packaging or do not have any deeper knowledge about a certain project will not have direct commit permissions. However, they can still create a copy of any package source and ask back to merge their changes. Open Build Service has support to make this easy.

Wiki reference: User comment page (http://en.opensuse.org/openSUSE:Build_Service_Collaboration)

3 How to integrate external SCM sources

Application development usualy happens in SCM systems like git, subversion, mercurial and alike. These external sources can be used directly in OBS via source services. OBS will always keep a copy of the sources to guarantee that the build sources are still available even when the external SCM server disappears or get altered.

3.1 How to create a source service

Let OBS create a tar ball out of an SCM repository. This just creates or extend a _service file with some rules how to download and package sources. The actual work happens on a local build or on a service side build. Please note that you need the obs-service-obs_scm installed for local runs.

```
# osc add https://SOME_URL.git
```

The web interface is creating as well a _service file when adding an URL to a SCM system.

3.1.1 Follow upstream branches

The created _service file is set up to follow latest source submissions on each run and looks like this:

```
<services>
  <service name="obs_scm">
    <param name="url">https://github.com/FreeCAD/FreeCAD.git</param>
    <param name="scm">git</param>
  </service>

  <service name="set_version" mode="buildtime"/>
  <service name="tar" mode="buildtime"/>
  <service name="recompress" mode="buildtime">
    <param name="file">*.tar</param>
    <param name="compression">xz</param>
  </service>
</services>
```

This will create an obscpio archive via the obs_scm service with the latest sources. This archive will get extracted at build time and be processed via the other services to build a compressed tar ball for rpmbuild. To follow a specific branch and additional parameter for "revision" is needed for the obs_scm service.

3.1.2 Fixed versions

You may want to build an archive for a fixed version, for example an official release which has been tagged by the upstream project. It is recommend to specify the mode="disabled" and to submit the archive via the following

```
# osc service runall
# osc ar
# osc commit
```

commands.

3.1.3 Avoid tar balls

tar balls are not a requirement by OBS, but by the packaging tool, eg. rpmbuild. However, you may want to decide not to ship a tar ball inside of the src.rpm. This makes sense for large sources where the compression time and needed disk space is just considered a waste for short living builds. You can simplify your _service file in that case, but you need to help rpmbuild to work directly in the source. It is also a good practice to package the _service file instead of the tar ball to give the user a chance to rebuild the src.rpm as long the external SCM server is providing the sources. The simplified _service file looks like this:

```
<services>
  <service name="obs_scm">
    <param name="url">https://github.com/FreeCAD/FreeCAD.git</param>
    <param name="scm">git</param>
  </service>

  <service name="set_version" mode="buildtime"/>
</services>
```

The spec file needs some hints to build inside the extracted sources directly. The macro can be used to switch to build tar balls or not to keep it working for stable releases where you want to provide a complete source rpm.

```
...
%define build_tar_ball 0
...
%if %{build_tar_ball}
Source0:        %{name}-%version.tar.xz
%else
Source0:        _service
%endif
...
%prep
%if %{build_tar_ball}
 %setup -q
%else
 %setup -q -n %_sourcedir/%name-%version -T -D
%endif
```

4 Publishing Upstream Binaries

This chapter covers main step of using OBS to publish binaries of your project for multiple distributions.

4.1 Which Instance to Use?

4.1.1 Private OBS Instance

OBS is open source project and therefore you can set up your own instance and run it by your own. The main advantage of this approach is that you can keep all your sources and build recipes unpublished if you need to (for example because of NDA). Obvious downside of this approach is that you need to maintain your own server/servers for running builds, publishing and mirroring. Also making your project public may attract some potential contributors.

More information about setting up your own private OBS instance can be found in *Chapter 5, Setting Up a Local OBS Instance* .

4.1.2 openSUSE Build Service

Other option is to use some publicly available instance of OBS. One good example is openSUSE Build Service at http://build.opensuse.org. This OBS instance can be used by anybody to freely create binaries for any of the supported distributions. Big advantage is that somebody is already taking care of all the infrastructure. You can store your sources there, build your packages and got them mirrored around the world. You do not need to get your own server and configure it, you can start using it right away.

4.2 Where to Place Your Project

This part helps you to decide on how to name and where to place your project and what project structure to create. This is more important if you are sharing your OBS instance with other people like in openSUSE Build Service (http://build.opensuse.org).

4.2.1 Base Project

If there are more packages in OBS, like for example in openSUSE Build Service (http://build.opensuse.org), these packages needs to be somehow divided into projects so it is easier to find what people are looking for and it is not all just one big mess.

In openSUSE Build Service, packages are divided into categories regarding their function. MySQL is in *server:database* repository, lighttpd in *server:http* and for example KMyMoney has its own subproject in *KDE:Apps*. So it is a good idea to think about in what category available on the OBS your application will fit the best.

If you need whole project for yourself - for example some of your dependencies is being built in the same project, you need to request creating subproject. In openSUSE Build Service, this is done through asking OBS admins for it on opensuse-buildservice mailing list (mailto:opensuse-buildservice+subscribe@opensuse.org). Its archive and link for subscribing can be found at http://lists.opensuse.org/opensuse-buildservice/ .

If you need to just put your package somewhere, you can create it in your home project and then send submitrequest to the project you want your package to get included in.

4.2.2 Supporting Additional Versions

If you want to support more than one version of your program, you need to use several projects. The same package cannot be contained in the same project multiple times.

4.2.2.1 Stable and Development Versions

Let's assume that you have found project suitable for your program. Some projects already have something like *STABLE* and *UNSTABLE* subprojects. So you can use these, if you discuss it with maintainers of these project. Other way is to ask somebody from the maintainers of the project to create either these subprojects (if they do not exist) or something similar. Always try to discuss it with the maintainers of the project. They might have good ideas, suggestions and may help you in various ways.

4.2.2.2 Multiple Stable Versions

If you want to support multiple version, you would need more projects than just two as suggested in previous section. These special projects should contain versions they are supposed to support in their name. If you are creating them under some project you are sharing with other packages, having you package name in the name of projects is a good idea as well.

GNOME is a good example: There is the *GNOME* project and many subprojects. Among them are, for example, *GNOME:STABLE:2.30*, *GNOME:STABLE:2.32*, and *GNOME:STABLE:3.0*. These projects hold different stable versions of GNOME with latest fixes.

4.3 Creating a Package

Packaging is quite a complex topic. Instead of trying to cover it in this book, it is a good idea to start with available internet documentation. One of the recommended online resource is Portal:Packaging on openSUSE wiki. You can find it at http://en.opensuse.org/Portal:Packaging. It contains links to several packaging tutorials and other packaging related documentation.

4.4 Getting Binaries

Note

The following sections discuss feature available only in openSUSE Build Service—a freely available instances of OBS.

For a nice download page for your software published on openSUSE Build Service, use the openSUSE download page. You can include it for example using either `iframe` or `object` on newer websites. An example of download page can be following one http://software.opensuse.org/download.html?project=openSUSE:Tools&package=osc. You can see how it looks like in *Figure 4.1, "openSUSE download page for package from OBS"*. It contains links to the packages and instructions how to install them.

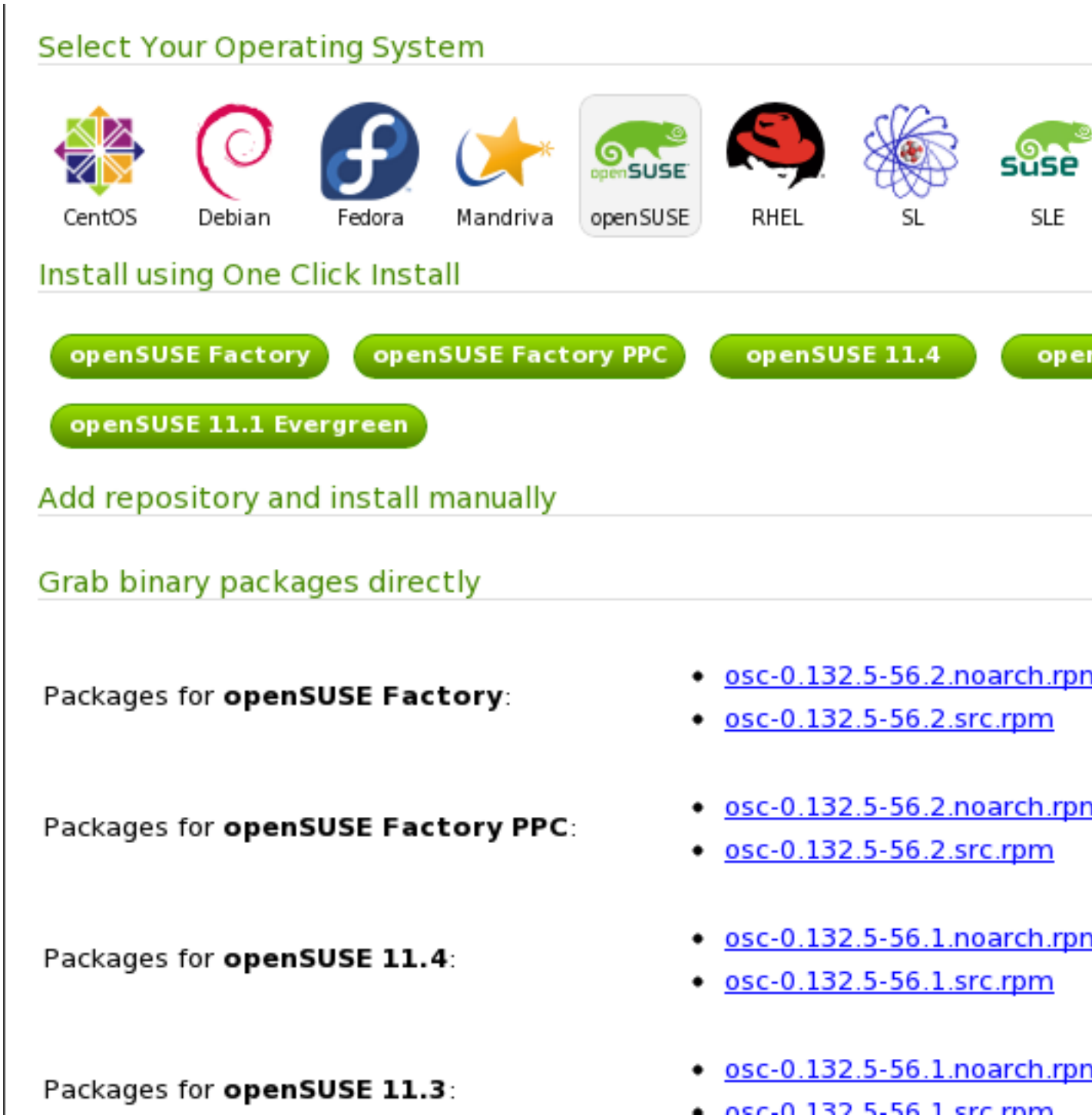

FIGURE 4.1: OPENSUSE DOWNLOAD PAGE FOR PACKAGE FROM OBS

URL always has to start with http://software.opensuse.org/download.html?.You can attach any number of &-separated parameters. But at least two of them - *project* and *package* - are required. All parameters with descriptions can be found in *Table 4.1, "Parameters for Download Page"*.

TABLE 4.1: PARAMETERS FOR DOWNLOAD PAGE

parameter	description
`project`	Project in which your package is located
`package`	Name of your package as it is specified in OBS
`bcolor`	Background color in hexa (for example `bcolor=004`) to make the download page better match your project page
`fcolor`	Text color in hexa (for example `fcolor=fff`) to make the download page better match your project page
`acolor`	Link color in hexa (for example `acolor=ff0`) to make the download page better match your project page
`hcolor`	Highlight color in hexa (for example `hcolor=0ff`) to make the download page better match your project page

4.4.1 Examples

Now we will take a look at how to include the download page into your project pages. As an example, we will use the **`osc`** client from the *openSUSE:Tools* project. To demonstrate the colors change, we will use theme that would match Midnight Commander.

First we will start with recent web page supporting new standards. On such a website, we would use `object` to include download code:

```
<object type="text/html"
   data="http://software.opensuse.org/download.html?
project=openSUSE:Tools&amp;package=osc&amp;bcolor=004&amp;acolor=ff0&amp;fcolor=fff&amp;hcolor=0ff"
   width="100%" height="100%">
   <param name="src"
      value="http://software.opensuse.org/download.html?
project=openSUSE:Tools&amp;package=osc&amp;bcolor=004&amp;acolor=ff0&amp;fcolor=fff&amp;hcolor=0ff" /
>
   Your browser doesn't support objects, please continue to the
```

```
   <a href="http://software.opensuse.org/download.html?
project=openSUSE:Tools&amp;package=osc&amp;bcolor=004&amp;acolor=ff0&amp;fcolor=fff&amp;hcolor=0ff">
   download page</a>
</object>
```

If you are using php on your server, you can make it easier by using following code:

```
<?php
   $url = "http://software.opensuse.org/download.html?
project=openSUSE:Tools&amp;package=osc&amp;bcolor=004&amp;acolor=ff0&amp;fcolor=fff&amp;hcolor=0ff";
   echo '
<object type="text/html"
   data="' . $url . '"
   width="100%" height="100%">
   <param name="src"
      value="' . $url . '" />
   Your browser doesn't support objects, please continue to the
   <a href="' . $url . '">download page</a>
</object>
';
?>
```

If you are running some legacy website, you might have to use `iframe` :

```
<iframe src="http://software.opensuse.org/download.html?
project=openSUSE:Tools&amp;package=osc&amp;bcolor=004&amp;acolor=ff0&amp;fcolor=fff&amp;hcolor=0ff"/
>
```

5 Setting Up a Local OBS Instance

This chapter explains how you could setup/Install/test OBS in your system. This chapter is written for those who are not so familiar with Linux and OBS. So in case you are confident to set up OBS, skip this chapter. Following would be explained in this chapter.

- OBS 1 click install, then manual setup in openSUSE 13.1;
- OBS 1 click install, then manual setup in SLES11;
- OBS test run on Microsoft Windows using VMware player;
- OBS appliance installed manually in a VirtualBox.

Unfortunately, didn't have a chance to install OBS in other Linux distribution yet. The last section will explain your first steps with the new OBS server.

How you could install and purchase openSUSE 13.1 and SLES12 will not be explained in this chapter. VMware player install and purchase also will not be explained. For these topics, you could visit for help:

- http://software.opensuse.org/131/en
- http://www.suse.com/products/server/
- http://my.vmware.com/web/vmware/free#desktop_end_user_computing/vmware_player/6_0

5.1 Testing OBS on Microsoft Windows Using VMware Player

Those who are not familiar with Linux can run and test OBS. To run and test OBS in Windows, you could use a virtual machine program such as VMware or VirtualBox, etc. This chapter explains, how you can run OBS using VMware player. To check and test with VirtualBox or another virtual machine, check in below.

After you are done installing VMware player on Windows, you need to download the OBS appliance program. You could get OBS appliance file by visiting http://openbuildservice.org/download/other/ and clicking on *VirtualBox/VMware Image*. After downloading, uncompress with some Windows archiving program that understands the `.tar.bz2` file format.

Now, open VMware Player application and select *File* › *Open a Virtual Machine*, or you could press Ctrl–O directly. Open the decompressed virtual machine in *Open Virtual Machine* dialog box. Click on *Play virtual machine* icon or hyper link in VMware player.

At the Linux prompt, you can login using "root" as a login name and "opensuse" as a password. Now, OBS local instance should be already loaded and running in your system. To make sure that the OBS Web UI is successfully up and running, open the OBS Web UI.

To access OBS Web user interface, open your web browser and try the address `http://vm.ip.address`. You can check the virtual machine's IP address by using **`ifconfig`** Linux command. Now, you probably could see a screen like the one below in your window:

To login your local OBS instance, you could use default login name as "Admin" and password as "opensuse". Check if you could login properly by clicking *Login* on your local OBS instance Web UI.

5.2 OBS 1-Click Install on openSUSE 13.1

This section explains, how you can install OBS on top of openSUSE 13.1. Open your web browser and go to that URL http://download.opensuse.org/repositories/openSUSE:/Tools/openSUSE_13.1/OBS_Server.ymp

This YaST Meta Package file lets you install OBS by 1 click. In case you are using openSUSE 11.2 or below version, check if you could find 1 click install file in this URL http://en.opensuse.org/openSUSE:Build_Service_Installation_SUSE

After you click on the above URL, you would see *Opening OBS_Server.ymp* window, Select default selection which is *Open with YaST Meta Package Handler* and press *OK* button. Then *OBS_Server Installation - YaST* window will appear. Select *Next* button after that till you get successful OBS setup message.

TIPS : In case you didn't install libMagickCore.so.2 in openSUSE, you might face dependency warning. In that case, stop the OBS install by pressing *Abort* button in *OBS_Server Installation - YaST* window. Then search and install libMagickCore using a search engine like Google or others and repeat above processes. Then you will get an OBS setup message.

Now, OBS Server is installed in your openSUSE. To run OBS server, you need to work on several further steps. Open /usr/share/doc/packages/obs-server/README.SETUP file. To run OBS server, you need to follow each step of README.SETUP carefully based on your system situation. You can also find this `README.SETUP` file in our public git repository (https://github.com/openSUSE/

open-build-service/blob/master/dist/README.SETUP.md). If you are using our stable release, you might want to switch to the corresponding branch, eg. 2.7 (https://github.com/openSUSE/open-build-service/blob/2.7/dist/README.SETUP).

If you follow instructions from README.SETUP file, you should be able to run OBS server in your system. There are several tips that I would like to comment regarding README.SETUP file.

TIP #1 : Check *3.1 Initialize fresh Database*. It might be described as

```
# cd /srv/www/obs/api/
# sudo RAILS_ENV="production" rake db:setup
# sudo chown lighttpd.lighttpd log/*

# cd /srv/www/obs/webui/
# sudo RAILS_ENV="production" rake db:setup
# sudo chown lighttpd.lighttpd log/*
```

but it should be

```
# cd /srv/www/obs/api/
# sudo RAILS_ENV="production" rake db:migrate
# sudo chown lighttpd.lighttpd log/*

# cd /srv/www/obs/webui/
# sudo RAILS_ENV="production" rake db:migrate
# sudo chown lighttpd.lighttpd log/*
```

For more information, see https://en.opensuse.org/openSUSE:Build_Service_Installation_SUSE.

TIP #2 : If you get fail message because of "apisrv" in "6. Using osc with your local build service:", try apiurl instead.

You might face some issues during process to follow README.SETUP file. If you read and follow instructions carefully, you could run OBS local instance on your system successfully.

5.3 OBS 1-Click Install on SUSE Linux Enterprise Server 12

1 click install for OBS in SLES12. This section explains how you install OBS in SLES12. Installation of OBS in SLES12 is quite similar to OBS install method on openSUSE. I would skip same contents so you might need to have a look for prior section "OBS 1 click install on openSUSE 13.1".

Before you install OBS on SLES12, you need to download and install SLES12 SDK first. you could download SLES12 SDK in here http://download.novell.com.

After you download and install SLES12 SDK, We could use YMP file for OBS 1 click install like openSUSE 13.1. Open your web browser and go to this URL http://download.opensuse.org/repositories/openSUSE:/Tools/SLE_11/OBS_Server.ymp.

Like for openSUSE 11.3, I have referenced the previous from http://en.opensuse.org/openSUSE:Build_Service_Installation_SUSE.

After you clicked on that URL, you would see *Opening OBS_Server.ymp* window as described in the previous openSUSE 13.1 section. Press *OK* button and *Next* button after YaST2 window pops up. If YaST lets you know successful install of OBS server, then you are ready to activate and run OBS on SLES12.

To run OBS server on SLES12, you need to follow instructions based on /usr/share/doc/packages/obs-server/README.SETUP file as described in prior section. From here, you could refer to prior section for successful OBS run on your SLES12.

5.4 Installing a Readymade OBS Appliance in a VirtualBox

This method is slightly less easy than the method using the readymade vmdk VMware disk, but it enables you to determine the size of your virtual disks to your convenience. It could also work with a real computer with two disks. It requires some knowledge of command line and partitioning.

1. Download the OBS appliance installer. Visit: http://www.openbuildservice.org/download and press the *Download the OBS Appliance Installer* button. It will start downloading an ISO image.
2. In VirtualBox, create a virtual machine with:
   - 4 GB memory
   - 1 virtual hard disk of 20 G for `/` and `/var/cache/obs`
   - 1 virtual hard disk of 50 G for `/srv/obs`

- a virtual CD-ROM driver pointing to the downloaded ISO image
- network bridging with real Ethernet card

3. Boot the virtual machine and choose to install the OBS server on the smaller virtual hard disk.
4. Log into the virtual machine with Login: root and Password: opensuse. If needed, switch to German/French/whatever keyboard: **`# loadkeys de`**. Inspect partitioning: **`# df -h`**. It shows you that the root partition is small and already almost full (1.6 GB used out of 1.8). Let's prepare the other partitions to get a bit more working space. First, **`# fdisk /dev/sda`** and prepare `/dev/sda2` to use the remaining space. Second, **`# fdisk /dev/sdb`** and prepare `/dev/sdb1` to use all the space, with type 8e (Linux LVM):

```
# pvcreate /dev/sdb1
# vgcreate OBS /dev/sdb1
# lvcreate -n server -L 48G OBS
# mkfs.ext4 /dev/OBS/server
```

5. Reboot, this time onto the hard disk. The CD-ROM might be disconnected, we will not need it anymore. Log in as root user, change keyboard if needed, and format `/dev/sda2`: **`# mkfs.ext4 /dev/sda2`**. Add following entry to /etc/fstab:

```
/dev/sda2  /var/cache/obs  ext4  defaults  2 1
```

Mount the new partition: **`# mount /dev/sda2`**. Get your IP address: **`# ifconfig`**.

5.5 First Steps with Your New OBS Server

At this point, one of the methods above should have provided you with a running OBS instance. Let us get our first package building.

1. From a web browser, access the web interface: https://vm.ip.address/. Accept the self-signed certificate.
2. In the top right corner of the web interface, there is a *Log In* option. Use it to log in as: Admin opensuse.
3. Click on the *Configuration* button to give your server a name and a description. Click on the *Interconnect* option. Choose *openSUSE* as the remote repository where we will pick up the packages of the build environment. Log out of the web interface.

4. Use *Sign Up* option to create a regular user account (for example: hmustermann).
5. As this normal user, click on the *Home Project* option and create your home project (that would be: "home:hmustermann").
6. Go to this home project, and click on *Create package* to create your first package (let's say: "mypackage").
7. Go back to your home project, and click the *Repositories* button. Choose to add a new repository and pick *openSUSE 13.1* (for example).
8. Reboot the virtual machine to ensure all projects are rescanned.
9. From outside the virtual machine and as a normal user, declare in `~/.oscrc` your new OBS user:

   ```
   [https://vm.ip.address]
   user=hmustermann
   pass=bond007
   ```

   then checkout your new package: **`$ osc -A https://vm.ip.address co home:hmustermann`**. go to your first package: **`$ cd home:hmustermann/mypackage`**. and add some sources there (tarball, spec file, changelog, patches). Check them in, then trigger a remote build:

   ```
   $ osc add *
   $ osc commit
   $ osc rebuild
   ```

10. The built packages can be seen at: http://vm.ip.address:82/

6 Bootstrapping

This chapter explains Boot strapping. In this chapter, You would learn how you could have other OBS projects and packages to your local OBS instance after your OBS install. There are some useful OSC commands examples and OBS working mechanism explanation in this chapter also. Basically this chapter is a copy from Build Service portal. For information about OBS bootstrapping on the Build Service portal, see https://en.opensuse.org/openSUSE:Build_Service_private_instance_boot_strapping.

6.1 The Issue

If you create a private instance of an OBS it is likely to be fully independent. This means that your OBS needs to build its full reference tool chain. This process—called bootstrapping—presents the same problem as the Chicken and the Egg, which one came first! In other words, you need to create a tool chain with the tool chain that you want to create.

6.2 A Cheat Sheet

6.2.1 Creating Your First Project

Log on to the Web API. The default user Admin, with the password `opensuse` is available. Create your own login and password and set yourself as Admin. Log on to the Web UI as Admin and click on the icon *Configuration* and add the openSUSE Build Service as the remote instance. Select from under *Locations* › *Projects*. At the end of the list, click *Add Project*. Give it a name (e.g. Meego-test) Select your new project and create a sub-project 0.1. You have now a project Meego-test:1.0 Sub projects are handy to propagate Access Control Lists (acl) and for creating the version as a sub project simplifies the user and project administration.

6.2.2 Importing Your Base Linux Project

We are now going to import the base project. I will describe two methods, one where you have a login on a remote OBS instance, one where you have only access to the rpm repository. In both cases you will need access to binary and source rpm.

6.2.2.1 With a login on a remote OBS

The osc copypac (I assume that you have installed the osc package on your workstation) has an option -t which enables copying towards a remote target OBS instance. osc help and osc help command will advise on how to use these. First you need to import the project configuration.

```
$ export PROJECT=MeeGo-test:0.1
$ osc -A http://api-url-source-obs meta prjconf $PROJECT > my_project.conf
$ osc -A http://api-url-target-obs meta prjconf -F my_project.conf $PROJECT
```

Then import the project. As you might have some Links in the project that you import, it is a good idea to keep the source and target project names identical.

```
$ PRJ=ProjectToCopy; for i in `osc -A http://api.source.obs.domain ls $PRJ`; do \\
osc -A http://api.source.obs.domain \\
copypac -t http://api.target.obs.domain $PRJ $i $PRJ ;done
```

6.2.2.2 Without a Login on a Remote OBS

If you have access only to the repositories of your source reference target, then your life will be a bit more difficult. My advice would be to recheck if you find you cannot get a login on a public OBS service - such as provided by openSUSE or MeeGo - before proceeding this way. You will not be able to import the project config and you will have to create it by hand. This is too long to be covered in this HowTo. For more explanation about Build Service project config, see http://en.opensuse.org/openSUSE:Build_Service_prjconf.

Then you need to download all your rpm source on to a local machine and import it into your project with the command.

```
$ osc importsrcpkg
```

6.2.2.3 Bootstrapping

To initiate the build process, we will copy the rpm binary from the source OBS of the source project. These binary rpms, from which we will remove any reference to release and version, will be used to trigger the first build. The OBS appliance will recompile all the rpms until all rpms in the project have been compiled only with packages compiled from their source code. Some base packages (e.g. tool chains) will be compiled several times during that process. Alternative You can build a first time against a target which is similar to the base that you need to boot

strap in lieu of building against your own base and change the build reference to your bootstrap base once that the first build has been successful. Remember that you can also build against remote baseline. Double check that the preliminary step have been executed correctly. You must have already : copied a Linux base distribution in an OBS project defined a build target for that base project.

If you have not defined a build target, the necessary directory structure will not exist. This is a mandatory step of preparation. Stop the scheduler as it will create a mess if the system is not stable:

```
# rcobsscheduler stop
```

* Add binaries to the :full directory of the Project ssh onto the OBS server. Now go to the project's build directory, and create a directory called ":full". Note : standard is the default name of your Build repository as defined in your project. It might change depending on who created the initial build repo.

```
# cd /obs/build/$PROJECT/standard/i586
```

This directory structure should already exist. If not, there is a problem (note that /obs is link and the target may vary with your implementation). Now create the ":full" directory. $ mkdir :full Copy over all the binary rpms of the project you are trying to build from scratch. These rpms should have the release and version numbers stripped from them. e.g. alsa-utils-1.0.22-2.7.i586.rpm -- should be – alsa-utils.rpm Note : If the original project has a :full directory you can copy from there to avoid the issue of stripping version and release numbers. * Add binaries to the :full directory of the Project. Change all user/group privileges under /srv/obs/build/ to "obsrun"

```
# chown -R obsrun:obsrun /srv/obs/build
```

If you leave root as owner of :full, it will still build but the scheduler will fail (almost silently) to upgrade :full with the latest built packages. Except in very special cases, it is very unlikely that you want to do so. * Start the OBS scheduler

```
# rcobsscheduler start
```

* Force the obs to reindex your new :full directory. It will send an event to the scheduler which will create a file named :full.solv

```
# obs_admin --rescan-repository $OBS-PROJECT $REPO $ARCH
```

6.2.2.3.1 Troubleshooting

At that time you should see your project restarting to build. If that would not be the case. * check that your files in your target :full directory are all own by the user obsrun. The following command should not return any file name.

```
#find /obs/build ! -user obsrun
#chown -R obsrun:obsrun  /obs/build   (will correct ownership issue)
```

* Force the obs to reindex your new :full directory. It will create a file named :full.solv

```
$obs_admin --rescan-repository $OBS-PROJECT $REPO $ARCH
```

* Check that your rpm are valid (e.g. not damaged during transfer)

```
#cd /obs/build/$PROJECT/standard/i586:full
#for I in `ls *.rpm` ; do rpm -qlp $I >/dev/null; if [ $? -ne 0 ] ;then echo $I >>../
error.lst ; fi ; done
#cat ../error.lst (must be empty, all rpm in error needs re-installation)
```

* Still not working, get a look in the log files in the directory /obs/log. You can start by having a look at /obs/log/scheduler_TARGET_ARCH.log and search from the end for the string "expanding dependencies". You will find from there why the scheduler fails.

```
#tail -f /obs/log/scheduler_i586.log
```

6.3 Creating a First Project

After creating a dedicated user via the Web API, log onto the Web UI again with your new login. Open your home project and create a sub project called "MyTest". To add a package in your new Home project, simply create a link [link Package from other Project] with one of the packages recently copied in your new OBS instance (see previous chapter Import your base project). Pick up a small one to speed compilation time. Click on the "+" near Build Repositories to add a repository. Move to the end of the page where all the standard Linux distributions are listed and click on [Advance]. Give a name to your repo, e.g. my-test and pick from the list the project/repo that you have just imported and rebuilt. This will request the OBS to build your new Home project against that repository. You can now check out your Home project with the osc command, modify a file or two and at your next check-in, the OBS will rebuild your Home project. If your reference project changes, the OBS will also rebuild your Home project.

7 `osc` Example Commands

This chapter explains and shows OSC commands examples. You could use OBS much more efficiently with OSC commands. $man OSC will show you [GLOBALOPTS], SUBCOMMAND, [OPTS][ARGS....]. You also could find some OSC commands examples from OBS Build Service portal. This chapter will take every OSC command examples from OBS Build Service portal and describes it in here. You could visit Build Service portal OSC command explanation at https://en.opensuse.org/Build_Service/CLI.

7.1 Package Tracking

With osc it is also possible to manage packages in a SVN like way. This feature is called package tracking and has to be enabled in ~/.oscrc's [general] section

```
# manage your packages in a svn like way
do_package_tracking = 1
```

Add a new package to a project

```
osc mkpac [package]
```

Add an already existing directory and its files to a project

```
osc add [directory]
```

Remove a package and its files from a project

```
osc deletepac [package]
```

All the commands above only change your local working copy. To submit your changes to the buildservice you have to commit them (osc ci -m [message]). The status command also displays the state of the packages

```
osc st
```

8 Advanced Project Setups

These best practices describe more complex setups, for example how to rebuild an entire stack with minimal effort.

8.1 Rebuilding an Entire Project with Changes

8.2 Integrating Source Handling

8.3 Using OBS for Automated QA

9 Building Kernel Modules

10 Common Questions and Solutions

This currently an unsorted list of asked questions.

10.1 Working with Limited Bandwidth

Packages can contain large files, esp. some tar balls can become quite large, in some real life examples several hundred mega bytes. This can be a problem when you need to work on the package via a slow connection.

10.1.1 Using the Web Interface

The web interface is the easiest way to edit simple things without the need of the checkout.

Disadvantages are

- Not the preferred solution for power packagers
- No local build possible
- Still a significant bandwidth is needed compared to the size of the edited file.

10.1.2 Using `osc` with Size Limit

`osc` offers to skip files with a certain size (specified with `-l` switch) on checkout. The limit is stored locally and you can also run an update later without downloading any large files. All other files can be edited, diffed and committed as usual.

Disadvantages are

- The checkout is incomplete
- No local build possible

10.1.3 Using `download_url`

Manage your large files via source services. The easiest way is to use `osc add $URL` which just stores a small `_service` file. The check will not contain the large files by default, but they get downloaded when needed via the service. However, they will never get committed, so this

is the best approach when you have a fast downstream, but slow upstream like with standard DSL connections. Also other users can trust your tar ball, esp. important when you do version upgrades on foreign packages.

Disadvantages are

- The generated files have the _service: prefix in check out (but not during build).

10.1.4 Using Source Services in `trylocal` Mode

Manage your large files via source services in try local mode for example with **`download_url`** or **`download_files`** service. This means you can be flexible depending on your current connection without changing the setup. The service is generating the file on the server side when you decide not to commit it, but you can also decide to commit it and avoid the _service: prefix on the files. Also other users can trust your tar ball, esp. important when you do version upgrades on foreign packages.

Disadvantages are

- A checkout may still need the size limit switch when last commit contained the large files.

A GNU Licenses

This appendix contains the GNU General Public License version 2 and the GNU Free Documentation License version 1.2.

GNU General Public License

Version 2, June 1991

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a). You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b). You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c). If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a). Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b). Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c). Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER

EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the “copyright” line and a pointer to where the full notice is found.

```
  one line to give the program’s name and
an idea of what it does.
  Copyright (C) yyyy name of author
```

```
  This program is free software; you can
redistribute it and/or
  modify it under the terms of the GNU
General Public License
  as published by the Free Software
Foundation; either version 2
  of the License, or (at your option) any
later version.
```

```
  This program is distributed in the hope
that it will be useful,
  but WITHOUT ANY WARRANTY; without even
the implied warranty of
  MERCHANTABILITY or FITNESS FOR A
PARTICULAR PURPOSE. See the
  GNU General Public License for more
details.
```

```
  You should have received a copy of the
GNU General Public License
  along with this program; if not, write
to the Free Software
  Foundation, Inc., 59 Temple Place -
Suite 330, Boston, MA 02111-1307, USA.
```

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

```
  Gnomovision version 69, Copyright (C)
year name of author
  Gnomovision comes with ABSOLUTELY NO
WARRANTY; for details
  type `show w’. This is free software,
and you are welcome
  to redistribute it under certain
conditions; type `show c’
  for details.
```

The hypothetical commands `show w’ and `show c’ should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w’ and `show c’; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a “copyright disclaimer” for the program, if necessary. Here is a sample; alter the names:

```
  Yoyodyne, Inc., hereby disclaims all
copyright
  interest in the program `Gnomovision’
  (which makes passes at compilers)
written
  by James Hacker.
```

```
  signature of Ty Coon, 1 April 1989
  Ty Coon, President of Vice
```

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License (http://www.fsf.org/licenses/lgpl.html)↗ instead of this License.

GNU Free Documentation License

Version 1.2, November 2002

Copyright (C) 2000,2001,2002 Free Software Foundation, Inc. 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document “free” in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of “copyleft”, which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The “Document”, below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as “you”. You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A “Modified Version” of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A “Secondary Section” is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document’s overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The “Invariant Sections” are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The “Cover Texts” are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A “Transparent” copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not “Transparent” is called “Opaque”.

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The “Title Page” means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, “Title Page” means the text near the most prominent appearance of the work’s title, preceding the beginning of the body of the text.

A section “Entitled XYZ” means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as “Acknowledgements”, “Dedications”, “Endorsements”, or “History”.) To “Preserve the Title” of such a section when you modify the Document means that it remains a section “Entitled XYZ” according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document’s license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.

C. State on the Title page the name of the publisher of the Modified Version, as the publisher.

D. Preserve all the copyright notices of the Document.

E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

H. Include an unaltered copy of this License.

I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.

N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.

O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties--for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See http://www.gnu.org/copyleft/.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License “or any later version” applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

```
  Copyright (c) YEAR YOUR NAME.
  Permission is granted to copy,
 distribute and/or modify this document
  under the terms of the GNU Free
 Documentation License, Version 1.2
  or any later version published by the
 Free Software Foundation;
  with no Invariant Sections, no Front-
Cover Texts, and no Back-Cover Texts.
  A copy of the license is included in
 the section entitled “GNU
  Free Documentation License”.
```

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the “with...Texts.” line with this:

```
  with the Invariant Sections being LIST
 THEIR TITLES, with the
  Front-Cover Texts being LIST, and with
 the Back-Cover Texts being LIST.
```

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.