

Open Build Serviceを 使ってみた話

各種ディストリビューション向け
パッケージングへの活用

Kentaro Hayashi

ClearCode Inc.

2023年4月 東京エリア・関西合同Debian勉強会





スライドはRabbit Slide Showにて公開済みです

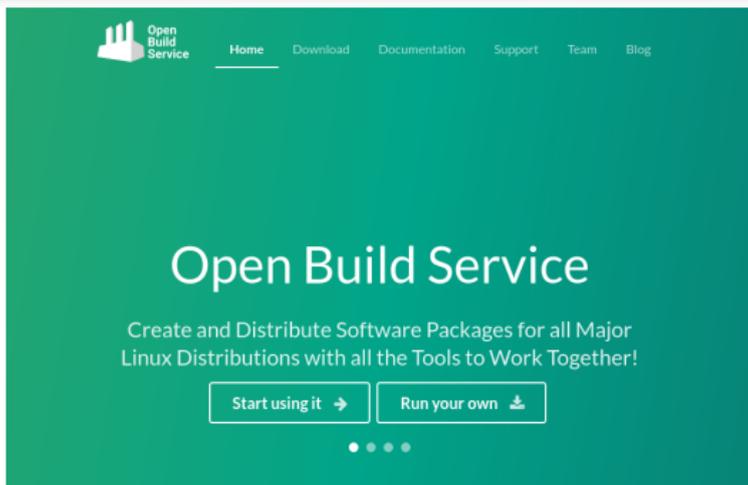
- Debianパッケージング Open Build Service編
 - <https://slide.rabbit-shocker.org/authors/kenhys/tokyodebian-obs-howto-202304/>



本日の内容

- Open Build Serviceを試す機会があったのでその紹介
 - どんなことができるのか？
 - どう使ったらいいのか？
 - どういう用途におすすめるのか？(個人の感想です)

Open Build Serviceとは



The screenshot shows the homepage of the Open Build Service website. The page has a teal background. At the top left is the logo, which consists of a stylized white building icon and the text "Open Build Service". To the right of the logo is a navigation menu with the following items: "Home" (underlined), "Download", "Documentation", "Support", "Team", and "Blog". In the center of the page, the text "Open Build Service" is displayed in a large, white, sans-serif font. Below this, a subtitle reads "Create and Distribute Software Packages for all Major Linux Distributions with all the Tools to Work Together!". At the bottom of the main content area, there are two white buttons with black text and icons: "Start using it →" and "Run your own ⬇️". Below the buttons are three small white dots, indicating a carousel or a sequence of slides.

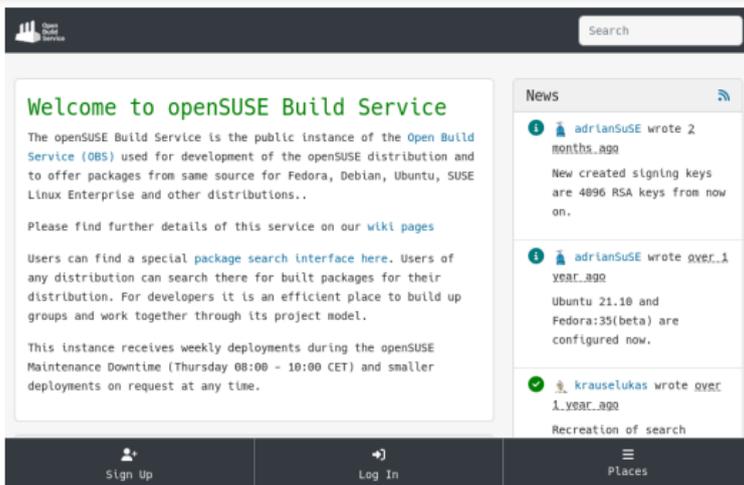
■ <https://openbuildservice.org/>



Open Build Serviceとは

- 公式サイトは <https://openbuildservice.org/>
- openSUSEの開発に利用されているソフトウェア
 - 幅広い環境向けにパッケージのビルドサービスを提供できるのが特徴
- OBSのインスタンスが <https://build.opensuse.org/>

build.opensuse.org



The screenshot shows the homepage of the openSUSE Build Service. At the top left is the logo for Open Build Service. A search bar is located at the top right. The main content area is divided into two columns. The left column contains a 'Welcome to openSUSE Build Service' message, explaining that the service is used for developing the openSUSE distribution and offering packages from various sources like Fedora, Debian, and Ubuntu. It also provides links to wiki pages and a package search interface. The right column is titled 'News' and lists three recent updates: two by 'adrianSUSE' regarding RSA keys and one by 'krauselukas' about a search recreation. At the bottom, there are three buttons: 'Sign Up', 'Log In', and 'Places'.

Welcome to openSUSE Build Service

The openSUSE Build Service is the public instance of the [Open Build Service \(OBS\)](#) used for development of the openSUSE distribution and to offer packages from same source for Fedora, Debian, Ubuntu, SUSE Linux Enterprise and other distributions..

Please find further details of this service on our [wiki pages](#)

Users can find a special [package search interface](#) here. Users of any distribution can search there for built packages for their distribution. For developers it is an efficient place to build up groups and work together through its project model.

This instance receives weekly deployments during the openSUSE Maintenance Downtime (Thursday 08:00 - 10:00 CET) and smaller deployments on request at any time.

News

- [adrianSUSE](#) wrote 2 months ago
New created signing keys are 4096 RSA keys from now on.
- [adrianSUSE](#) wrote over 1 year ago
ubuntu 21.10 and Fedora:35(beta) are configured now.
- [krauselukas](#) wrote over 1 year ago
Recreation of search

Sign Up Log In Places

• <https://build.opensuse.org/>



Open Build Serviceを知る には

- 第1回 Open Build Service道場
 - <https://www.slideshare.net/ftake/1-open-build-service>
 - 2012年の資料ではあるものの参考になるはず



Open Build Serviceを知る には

- Open Build Service道場
 - <https://www.slideshare.net/ftake/obs-dojoadv>
 - パッケージの新規作成編 **作りかけバージョン**
 - 2014年の資料



どんなことができるのか？

- debやrpmのビルドができる
 - Debian, Ubuntu, Fedora, openSUSE
- ビルドしたパッケージのリポジトリを公開できる



類似サービスとの違い

- packagecloud.io
- launchpad.net



類似サービスとの違い(1)

- <https://packagecloud.io>
 -  リポジトリのホスティングを提供
 -  パッケージのビルドは自前で行う必要がある



類似サービスとの違い(2)

- <https://launchpad.net>
 - リポジトリのホスティングを提供
 - パッケージのビルドもおまかせ
 - サポートされているUbuntuのリリースのみ



Open Build Service

- 便利なところは？
- 気になるところは？



便利なところ(1)

-  RPM系だけでなく、Debもサポートしている
-  ソースパッケージをアップロードすると、あとはおまかせにできる
 -  パッケージの署名用の鍵のメンテナンス不要



便利なところ(2)

- ミラーを利用できる
 - <https://en.opensuse.org/MirrorCache>
 - <https://mirrorcache.opensuse.org> を指定してあれば、適宜適切なミラーが自動的に選択される
 - 日本の場合は <https://mirrorcache-jp.opensuse.org>
- プロジェクトのroleを設定することで複数人でメンテナンスできる



便利なところ(3)

- ソースパッケージまで準備しておけばあとはoscコマンドを使って自動化できる
- バージョン名を付与したパッケージを作ることで複数バージョンを公開できる
- いわゆる独立したプロジェクトとしてメンテナンスすることもできる(はず) **要確認**

- 個人のリポジトリは home:(ユーザー名)となる。公式扱いだと science:(xxx) みたいにできる



気になるところ(1)

- ~~×~~ <https://mirrorcache-jp.opensuse.org> へのミラーはそれほど早くない
 - 1日くらいみておくとよい
 - RPM系だと既定では <https://download.opensuse.org> への.repoがダウンロードできるようになっているので、ミラーを明示的に指定した場合にその影響を受ける



気になるところ(2)

- **X** 独自プロジェクトにするには手間と時間がかかる
- **X** 未対応のディストリビューションもある
 - 例: AmazonLinux



OBSを利用するための準備

- 最初にすること
- home:プロジェクトの準備
- プロジェクトの設定
- プロジェクトのフラグ設定
- パッケージの設定
- 参考: https://en.opensuse.org/openSUSE:Build_Service_Tutorial



最初にすること

- <https://build.opensuse.org/> にアカウントを作成する
 - 既定では home:(ユーザー名) プロジェクトが利用できる
- osc コマンドを利用できるようにする
 - `sudo apt install -y osc`



home:(ユーザー名)プロジェクトの準備

- `osc checkout home:(ユーザー名)`でプロジェクトをチェックアウトする(SVNっぽい感じ)
 - 初回のチェックアウトでアカウント情報を入力することで`~/.config/osc/osrcrc`が作られる
 - `home:(ユーザー名)`というディレクトリがそのまま作成される



プロジェクトの設定

- プロジェクトのメタ情報
 - 対象ディストリビューションとひもづける

```
osc meta prj home:(ユーザー名) --file=prj-meta.xml
```



プロジェクトのメタ情報

- リポジトリ名と依存ディストリビューションを指定

```
<project name="home:(ユーザー名)">
<title/>
<description/>
<person userid="kenhys" role="maintainer"/>
<repository name="bullseye">
  <path project="Debian:11" repository="update"/>
  <path project="Debian:11" repository="standard"/>
  <arch>x86_64</arch>
  <arch>i586</arch>
  <arch>aarch64</arch>
</repository>
</project>
```



複数人でメンテナンスする 場合

- roleをプロジェクトのメタ情報に含める

```
<person userid="kenhys" role="maintainer"/>  
<person userid="kou" role="maintainer"/>
```

プロジェクトのメタ情報

The screenshot shows the OBS web interface for the project 'home:kenhys'. The page title is 'Meta Configuration of home:kenhys'. The configuration is displayed as XML code in a text editor with line numbers 1 through 14. The code defines the project name, title, description, maintainers, repository name, and paths for different architectures and repositories.

```
1 <project name="home:kenhys">
2   <title>repository try & error OBS feature</title>
3   <description>This repository is aimed to check OBS functionality.
4 </description>
5   <person userid="kenhys" role="maintainer"/>
6   <person userid="kou" role="maintainer"/>
7   <repository name="jammy">
8     <path project="Ubuntu:22.04" repository="update"/>
9     <path project="Ubuntu:22.04" repository="standard"/>
10    <path project="Ubuntu:22.04" repository="universe"/>
11    <path project="Ubuntu:22.04" repository="universe-update"/>
12  </arch>x86_64</arch>
13  </repository>
14  <repository name="focal">
```

- [https://build.opensuse.org/projects/home:\(ユーザー名\)/meta](https://build.opensuse.org/projects/home:(ユーザー名)/meta)



リポジトリ一覧

The screenshot shows the OBS web interface for user 'home:kenhys'. The page title is 'Repositories for home:kenhys'. There are four repository cards displayed in a 2x2 grid:

- almalinux8 x86_64**: Repository paths: AlmaLinux:8/standard, AlmaLinux:8/baseos, AlmaLinux:8/extras, More. Download icon.
- almalinux9 x86_64 aarch64**: Repository paths: AlmaLinux:9/baseos, AlmaLinux:9/standard, AlmaLinux:9/extras, More. Download icon.
- bullseye x86_64**: Repository paths: (partially visible)
- centos7 x86_64**: Repository paths: (partially visible)

At the bottom of the interface, there are three buttons: 'Sign Up', 'Log In', and 'Places'.

- [https://build.opensuse.org/repositories/home:\(ユーザー名\)](https://build.opensuse.org/repositories/home:(ユーザー名))



ディストリビューションの 対応

- どれくらいのディストリビューションに対応しているのか？

```
$ osc dists
```



プロジェクトのフラグ

- 特定のディストリビューションで必要
 - AlmaLinux:8などのようにAppStreamでモジュールとして各種パッケージが提供されている場合(dnf module listで対象を確認できる)、OBSでは既定でモジュールは有効になっていない
 - AppStreamかつモジュールのパッケージをspecのBuildRequires:に指定するとunresolvableとなってビルドすらされない。
 - nothing provides: xxxxとそっけないメッセージがでるだけ



特定のモジュールを有効にするには

- プロジェクトのフラグを設定する
 - 例: `osc meta prjconf home:kenhys -file prjconf.txt`

```
ExpandFlags: module:ruby-3.1
```

- 参考: <https://openbuildservice.org/help/manuals/obs-user-guide/cha.obs.prjconfig.html>



プロジェクトのフラグ

The screenshot shows the OBS web interface. At the top, there's a search bar and the text 'home:kenhys / Configuration / Project Config'. Below this is a navigation menu with tabs: Overview, Repositories, Monitor, Requests, Users, Subprojects, Project Config (selected), and Attributes. The main content area is titled 'Project Configuration of home:kenhys'. It features an 'Undo' button, a 'Redo' button, and a settings gear icon. The configuration text shows '1 ExpandFlags: module:ruby-3.1' and '2'. Below the text is a 'Describe your changes' input field and a 'Save' button. At the bottom, there are three icons: Watchlist, Notifications (with a '2' badge), and Places.

- [https://build.opensuse.org/projects/home:\(ユーザー名\)/prjconf](https://build.opensuse.org/projects/home:(ユーザー名)/prjconf)



パッケージごとにする

- パッケージのメタ情報の設定
 - 特定のディストリビューションに対応していない場合、パッケージごと無効にできる
 - プロジェクトのメタデータで設定したリポジトリ名を指定する
 - <build>のターゲットとして<disable>を指定する

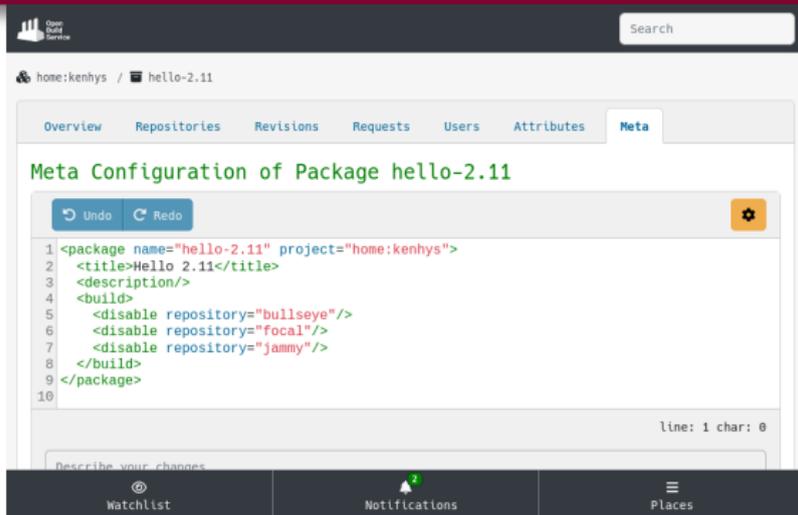
```
$ osc meta pkg hello-2.11 --file hello2.11-meta.xml
```



パッケージのメタ情報の例

```
<package name="hello-2.11" project="home:kenhys">  
  <title>Hello 2.11</title>  
  <description/>  
  <disable repository="bullseye"/>  
  <disable repository="focal"/>  
  <disable repository="jammy"/>  
</package>
```

パッケージのメタ情報の例



The screenshot shows the Open Build Service (OBS) web interface. At the top, there is a search bar and the user's home directory 'home:kenhys' and the package name 'hello-2.11'. Below this, there are tabs for 'Overview', 'Repositories', 'Revisions', 'Requests', 'Users', 'Attributes', and 'Meta'. The 'Meta' tab is selected, displaying the 'Meta Configuration of Package hello-2.11'. The configuration is shown in a code editor with the following XML content:

```
1 <package name="hello-2.11" project="home:kenhys">
2   <title>Hello 2.11</title>
3   <description/>
4   <build>
5     <disable repository="bullseye"/>
6     <disable repository="focal"/>
7     <disable repository="jammy"/>
8   </build>
9 </package>
```

At the bottom of the interface, there are buttons for 'Watchlist', 'Notifications' (with a '2' notification badge), and 'Places'.

- [https://build.opensuse.org/package/meta/home:\(ユーザー名\)/hello-2.11](https://build.opensuse.org/package/meta/home:(ユーザー名)/hello-2.11)



OBSでパッケージをビルド する方法

- rpm編
- deb編



.rpmのビルド方法

- SRPMから作成する方法
 - PackageCloudから移行するなら、SRPMをインポートするのがお手軽
- .specと関連するソースアーカイブから作成



SRPMを使う場合

- home:(ユーザー名) で次のコマンドを実行する

```
$ osc importsrcpkg -n hello-2.10 hello-2.10.src.rpm  
$ cd hello-2.10  
$ osc add hello.spec  
$ osc add hello-2.10.tar.gz  
$ osc commit -m "Add 2.10"
```



.specや各種ソースアーカイブを使う場合

- `osc meta`でパッケージに関するメタ情報を追加し、アーカイブを追加する

```
$ osc meta pkg home:(ユーザー名) hello-2.10 --file=hello2.10-meta.xml
$ osc update
$ cd hello-2.10
$ osc add (.specとか.tar.gzとか)
$ osc commit -m "Add 2.10"
```



パッケージのメタ情報設定

```
<package name="hello-2.10" project="home:kenhys">  
  <title>Hello 2.10</title>  
  <description>GNU Hello</description>  
</package>
```



.debのビルド方法

- .dsc, orig.tar.gz, debian.tar.xzを追加する

```
$ cd hello-2.10  
$ osc add hello_2.10-3.dsc  
$ osc add hello_2.10-3.debian.tar.xz  
$ osc add hello_2.10.orig.tar.gz  
$ osc commit -m "Add 2.10"
```

ビルド状態の確認

The screenshot displays the 'Build Results' page in the Open Build Service (OBS) interface. The page is organized into sections for different operating systems: a1n1linux8, a1n1linux9, and bullseye. Each section lists builds for various architectures (x86_64, aarch64) and provides a summary of their status: succeeded, excluded, finished, and disabled. A 'Refresh' button is visible in the top right corner. The bottom navigation bar includes 'Watchlist', 'Notifications', 'Actions', and 'Places'.

OS	Architecture	succeeded	excluded	finished	disabled
a1n1linux8	x86_64	9	4		
a1n1linux9	aarch64	9	4		
a1n1linux9	x86_64	9	4		
bullseye	x86_64	5	2	1	5

- osc resultsでも確認できる



リポジトリの利用方法

- .rpm
- .deb

例: AlmaLinux:8

Download Shortcuts ▾ Reports ▾ 88

^ / repositories / home: / kenhys / almalinux8

Show 20 ▾ entries Search:

Name	↑↓	Last Modified ↑↓	Size ↑↓
repdata/		2023/4/7 19:54	
src/		2023/4/7 19:54	
x86_64/		2023/4/7 19:54	
home:kenhys.repo		2023/4/7 19:54	251

Showing 1 to 4 of 4 entries Previous 1 Next

© 2021,2022 SUSE LLC., openSUSE contributors [Legal notice](#) [Source code](#) Hosting sponsored by
William Brown
. Trans Rights are Human Rights.
[Report issue](#)
MirrorCache 1.058

■ .repoが公開されているのでそれを利用

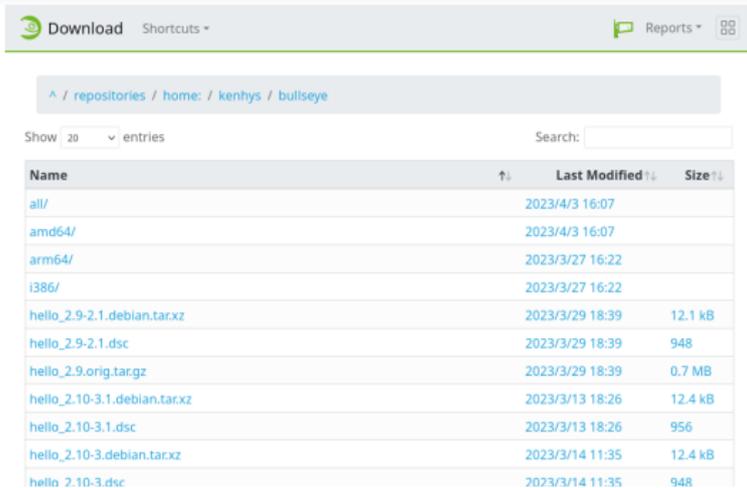


rpmの場合

- /etc/yum.repo.dに配置する

```
$ cd /etc/yum.repo.d  
$ sudo curl -L -O https://mirrorcache-jp.opensuse.org/  
repositories/home:/(ユーザー名)/(ディストリビューション)/home:(ユーザー名).repo
```

例: Bullseye



The screenshot shows a web interface for a repository. At the top, there is a navigation bar with 'Download' and 'Shortcuts'. Below that, a breadcrumb path reads '^ / repositories / home: / kenhys / bullseye'. There is a 'Show' dropdown set to '20' and a 'Search:' input field. The main content is a table listing files and directories.

Name	↑↓	Last Modified ↑↓	Size ↑↓
all/		2023/4/3 16:07	
amd64/		2023/4/3 16:07	
arm64/		2023/3/27 16:22	
i386/		2023/3/27 16:22	
hello_2.9-2.1.debian.tar.xz		2023/3/29 18:39	12.1 kB
hello_2.9-2.1.dsc		2023/3/29 18:39	948
hello_2.9.orig.tar.gz		2023/3/29 18:39	0.7 MB
hello_2.10-3.1.debian.tar.xz		2023/3/13 18:26	12.4 kB
hello_2.10-3.1.dsc		2023/3/13 18:26	956
hello_2.10-3.debian.tar.xz		2023/3/14 11:35	12.4 kB
hello_2.10-3.dsc		2023/3/14 11:35	948

- Release.keyが公開されているのでそれを利用



debの場合

- Release.keyが公開されているので、keyringに変換する

```
$ curl -L -O \  
https://mirrorcache-jp.opensuse.org/repositories/home:/(ユーザー名)/bullseye/Release.key  
$ gpg --no-default-keyring --keyring ./archive-hello-keyring.gpg --import Release.key  
$ sudo mv archive-hello-keyring.gpg /usr/share/keyrings/
```



debの場合(2)

- .listを追加してリポジトリへの参照を追加する

```
# cat /etc/apt/sources.list.d/hello.list  
deb [signed-by=/usr/share/keyrings/archive-hello-keyring.gpg] \  
http://mirrorcache-jp.opensuse.org/repositories/home:(ユーザー名)/(ディストリビューション)/
```



ハマりポイント(1)

- OBSでパッケージのビルドがはじまらない
 -  メタデータに対象リポジトリが不足していると失敗します
 -  不足しているメタデータを追加する
 -  不足しているプロジェクトのフラグを追加する



ハマりポイント(2)

- DockerではビルドできるのにOBSではビルドできない
 -  rootユーザーでのビルドを前提していると失敗します
 -  一般ユーザーでビルドできるように修正する



ハマりポイント(3)

- 複数バージョンを公開したいがビルドされなくなった
 -  バージョンつけずに複数ソースをコミットした
 -  バージョンつけてパッケージを管理する
- 例: helloディレクトリ配下に複数バージョンのソースをコミットすると正常にパッケージがビルドされなくなる



ハマりポイント(4)

- homeじゃないプロジェクトの作成難易度が高い
 -  <https://build.opensuse.org/project/new> で新規プロジェクトを作成できる
 -  home:配下しか作成できない
 - science:とかserver:mailなどのカテゴリがあるが、あくまでそちらはopenSUSEの開発で使われるプロジェクト向け
 - 他の組織がビルドサービスを利用する場合にはisv:配下にプロジェクトを作るのがおすすめ



homeじゃないプロジェクトの作成難易度が高い問題

- **X** <https://build.opensuse.org/project/new> では上記に該当しないプロジェクトを作成する場合にはメールで連絡をとるように案内されているが、その連絡先は noreply@opensuse.org となっており、メールを受け付けるつもりがない
- **X** 既知の問題 <https://github.com/openSUSE/open-build-service/issues/12356> だが修正されない



homeじゃないプロジェクトの作成難易度が高い問題

- admin@opensuse.org に連絡すると <https://progress.opensuse.org/> にissueを作ってもらえたりするが `opensuse-admin-obs` にアサインしてもらえないと気づかれない
- OBSとはチームが別なので気づいてもらえるの待ち



どんな用途に向くのか？

- 自分が書いたソフトウェアを様々なディストリビューション向けに一括で提供したい
- バージョンアップの頻度が高く各ディストリビューションのお作法にならってパッケージングがしんどい
- パッケージリポジトリの維持管理に手間をかけられない
 - 自分がupstream authorなソフトウェアならisv:配下にプロジェクトを作成するのがおすすめ
 - プロジェクト作成までは時間がかかるのは覚悟すべし