



Grand Unified Debian



銀河系唯一のDebian専門誌

東京エリア/関西Debian勉強会



あんどきゅめんとてっど でびあん 2023年冬号 2023年12月31日 初版発行

今更な勉強会

目次

1	Introduction	2
2	Debian Update	3
3	Debian で PHP が動く Web サーバを構築してみる (apache2 / nginx)	10
4	DDTP 及び DDTSS の紹介	18
5	Debian パッケージング Open Build Service 編	20
6	Debian パッケージング アップロード編	28
7	Analysis of the Debian Developer Survey (2022)	34
8	2022 年を振り返って	36

1 Introduction

DebianJP



1.1 東京エリア Debian 勉強会

Debian 勉強会へようこそ。これから Debian の世界にあしを踏み入れるという方も、すでにどっぷりとつかっているという方も、月に一回 Debian について語りませんか？

Debian 勉強会の目的は下記です。

- Debian Developer (開発者) の育成。
- 日本語での“開発に関する情報”を整理してまとめ、アップデートする。
- 場の提供。
 - 普段ばらばらな場所にいる人々が face-to-face で出会える場を提供する。
 - Debian のためになることを語る場を提供する。
 - Debian について語る場を提供する。

Debian の勉強会ということで究極的には参加者全員が Debian Package をがりがりとするスーパーハッカーになった姿を妄想しています。情報の共有・活用を通して Debian の今後の能動的な展開への土台として、“場”としての空間を提供するのが目的です。

1.2 関西 Debian 勉強会

関西 Debian 勉強会は Debian GNU/Linux のさまざまなトピック (新しいパッケージ、Debian 特有の機能の仕組、Debian 界隈で起こった出来事、などなど) について話し合う会です。

目的として次の三つを考えています。

- ML や掲示板ではなく、直接顔を合わせる事での情報交換の促進
- 定期的に集まれる場所
- 資料の作成

それでは、楽しい一時をお楽しみ下さい。

2 Debian Update

dictoss さん



本記事は元原稿の作成された 2023 年 3 月現在の情報をもとに記載しています。

2.1 Debian とは？

Debian とは？

ボランティアでフリー/オープンでユニバーサルな「オペレーティングシステム (OS)」を開発する「コミュニティ」

<https://www.debian.org/>

Debian はボランティアのみで開発しています

ディストリ	企業	ボランティア
Fedora	Red Hat 支援あり	あり
CentOS Stream	Red Hat 支援あり	あり
RHEL* ¹	Red Hat	なし
Debian	なし	あり
Ubuntu	Canonical	あり
openSUSE	SUSE 支援あり	あり
SLES* ²	SUSE	なし

- 様々な用途に使える汎用的な作り
 - ノート PC、デスクトップ PC などの普段利用するコンピュータの OS
 - Linux サーバの OS
 - * web サーバのシェア (2023-03-11 時点)*³ :
Linux サーバの内、Ubuntu 31.9 %、Debian 16.8 %、CentOS 8.4 %
 - 組込デバイスのベース OS (多くの CPU で動作する)
- 「Debian」ベースな派生 OS の源流
 - 例 : Ubuntu、Kali Linux、Raspberry Pi OS
 - 派生先のディストリビューションと相互に情報交換をして開発している
- 2023 年 3 月の時点で、最新の安定版は Debian 11.6
 - コードネーム: bullseye
 - パッケージ数は 約 59,551 以上を提供
 - 公式にサポートする CPU アーキテクチャは 9 つ

*³ <https://w3techs.com/technologies/details/os-linux/all/all>

- コードネームはトイ・ストーリーのキャラクター名を採用
- 次のメジャーリリースは Debian 12
 - コードネーム: bookworm
 - 2023 年夏頃にリリースすると思われる

2.2 リリースサイクル

- メジャーリリース: おおよそ 2 年ごと
 - Debian は time-based freeze を採用
- 標準サポート: 3 年
- LTS^{*4}: 標準サポート終了後から 2 年
 - 対象アーキテクチャは amd64、i386、arm 系に限定
 - 主要なパッケージのみサポート (全パッケージではない)
- Extended LTS: LTS 終了後から 5 年
 - 公式の Debian プロジェクトの運営ではなく、LTS をメンテナンスしている Freexian の商用サービス
 - 契約者がサポートするパッケージを指定する
 - 更新やセキュリティ修正したパッケージはすべての Debian ユーザーが無料で利用可能
 - お願い: できるだけ次のバージョンに上げてください

2.3 Debian と コミュニティ

Debian への参加者は世界中にいます

- Debian 開発者 (Debian Developer、DD)
 - Debian Project の公式開発者
 - 57 ヶ国に 939 名 (2023-03-11 時点)^{*5}
 - 日本に在住している人は 34 人
 - 国籍人数ではドイツ、アメリカ、フランス、イギリスに次いで第 5 位
- Debian メンテナー (Debian Maintainer、DM)
- パッケージメンテナー
- ドキュメントなどの翻訳
- そのほかにも多くの貢献者たちが Debian に参加
- Debian 社会契約^{*6}
 - Debian 開発者 たちが目指すフリーソフトウェアコミュニティの在り方
- Debian フリーソフトウェアガイドライン (DFSG)
 - Debian 社会契約の一部
 - Debian が考えるフリーソフトウェアの定義
 - オープンソースの定義のひな形にもなっている
- Debian Policy^{*7}
 - Debian パッケージの区分、内容、ルール、ファイル配置の方針などの技術的な定義
- DebConf は年に一度、Debian 開発者が集まって開催するカンファレンス

^{*4} Long Term Support

^{*5} <https://people.debian.org/~eriberto/udd/dd-by-country.html>

^{*6} https://www.debian.org/social_contract

^{*7} <https://www.debian.org/doc/debian-policy/>

- 通常はオフラインで集まるが、COVID-19 の影響で 2020 年と 2021 年はオンラインで開催
 - 2019/07/21 - 07/28: Debconf19: CURITIBA, BRAZIL
 - 2020/08/23 - 08/29: Debconf20: Online
 - 2021/08/24 - 08/28: Debconf21: Online
 - 2022/07/17 - 07/24: Debconf22: Prizren, Kosovo
 - * <https://debconf22.debian.org/>
 - * 発表のビデオがありますのでぜひご覧ください
 - 2023/09/10 - 09/17 (予定): Debconf23: Kochi, India
 - * <https://debconf23.debian.org/>

2.4 まとめ

まとめると「Debian」とは

- フリー/オープンな「オペレーティングシステム (OS) 」を作成しようとする人たちが集まるボランティアベースの「プロジェクト」
- 自分たちの考えるフリーという言葉に関する定義、開発目的、パッケージングポリシーを厳格に決めている
- 世界中に 900 人以上の公式開発者があり、他のディストリビューションのベースとして採用されている
- 約 2 年毎にリリースが行われ、多くのパッケージとアーキテクチャをサポートしている
- 上記のような特徴から様々なところで利用されている Linux ディストリビューション

2.5 Debian JP Project と Debian 勉強会

2.5.1 Debian JP Project とは？

- <https://www.debian.or.jp/>
- 日本において Debian を普及させることを目的とした任意団体
- 活動内容
 - Debian の日本語による情報発信
 - ユーザとの情報交換
 - Debian 開発者やパッケージメンテナの育成など

2.5.2 Debian 勉強会とは？

- 2005 年 1 月開始
- Debian 開発者 上川さんが発起人
- 東京と関西で月に一回コンスタントに開催している Debian 開発者、Debian ユーザによる勉強会
 - 東京エリア Debian 勉強会
 - * <https://tokyodebian-team.pages.debian.net/>
 - 関西 Debian 勉強会
 - * <https://wiki.debian.org/KansaiDebianMeeting>
 - COVID-19 のため、2020 年 3 月からは東京エリアと関西でオンラインによる合同勉強会を開催中

2.5.3 Debian 勉強会: 解決したい内容

- 問題
 - ML と IRC で情報交換していた

- face-to-face で会う場所がない
- まとまったドキュメントが出てこない
- Debian 勉強会の提案
 - 定期的集まる
 - 資料を作成して公開 (GPL-2+)

<https://salsa.debian.org/tokyodebian-team/monthly-report>

2.5.4 最近の勉強会

- 勉強会の内容
 - Debian 界限やパッケージング関連の話題など専門の人に話を聞く
 - Debian で気になった事柄を調べてレポートする
- 前回の内容 (2月):
 - 場所: オンライン
 - セミナー「DDTP 及び DDTSS の紹介」
 - ハンズオン「DDTSS を使ってパッケージ説明文を翻訳してみよう」
- 各地のイベントで Debian 普及活動
 - OSC2021 東京

2.6 次期安定版 Debian 12 bookworm

2023 年 3 月現在開発を進めている次の安定版リリース Debian 12 (コードネーム : bookworm)
 追記:Debian 12 は 2023 年 6 月 10 日にリリースされました

- <https://wiki.debian.org/DebianBookworm>
- Debian は time-based freeze を採用しており、およそ 2 年毎のリリースを目指す
 - Debian 12 でサポートしているアーキテクチャ
 - * amd64, i386 (i686 以降)
 - * arm64, armhf, armel
 - * mips64el, mipsel
 - * ppc64el
 - * s390x

2.7 インストーラ

Debian Installer Bookworm Alpha 2 release

- <https://www.debian.org/devel/debian-installer/News/2023/20230219>
- いち早く bookworm を試してみたい方は以下からダウンロードしてインストールできます
 - <https://www.debian.org/devel/debian-installer/index.ja.html>
 - 自分の環境でインストールがうまく行われない場合はインストールレポートをお願いします
 - <https://www.debian.org/releases/bookworm/amd64/apas04.ja.html>

2.8 テーマ

テーマ

- 次期安定版に採用するテーマは通常コンテストを行う
- しかし、今回は応募が 3 つと少なかったためコンテストは行わなかった
- Juliette Taka さん提供の「Emerald」で異論が出なければ、これが採用予定
- <https://wiki.debian.org/DebianArt/Themes/Emerald>



2.9 提供ソフトウェアのバージョン

ソフトウェアは以下のバージョンでテスト中

- Linux カーネル 6.1
- ツールチェイン GCC 10.4.0, binutils 2.40, glibc 2.36, LLVM 14.0.6, 13.0.1
- Perl 5.36.0, Python 3.11.2, Ruby 3.1.2, PHP 8.2.2, Go 1.19.6, OpenJDK 17.0.6
- GNOME 43, KDE Plasma 5.27.0, Cinnamon 5.6.7, MATE 1.26, Xfce 4.18, lxde 0.99.2, lxqt 1.2.0
- MariaDB 10.11.2, PostgreSQL 15.2, sqlite 3.40.1
- OpenSSH 9.2p1, OpenSSL 3.0.8, GnuPG 2.2.40/1.4.23
- etc..

2.10 リリースまでの流れ

Debian 12 はいつリリースされるのか？

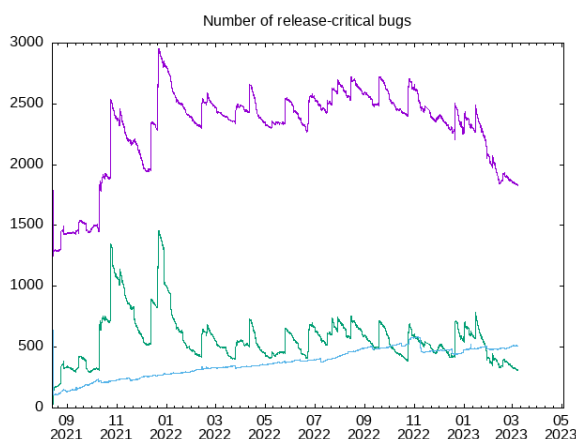
- Debian のリリース条件は「リリースクリティカルバグ (RC bug) が 0 個になったとき」
- 2023-03-08 時点の RC bug の数は「311」
 - <https://bugs.debian.org/release-critical/>

Release-critical bugs status

Wed Mar 8 02:00:00 UTC 2023

Total number of release-critical bugs: 1835
Number that have a patch: 195
Number that have a fix prepared and waiting to upload: 24
Number that are being ignored: 43
Number concerning the current stable release: 512
Number concerning the next release: 311

Number concerning the previous stable release: 42



リリースまでの流れ (フリーズスケジュール)*8

- 2023-01-12: Transition and Toolchain Freeze
 - コンパイラ等のビルドツールやプログラム実行環境などの重要なパッケージのバージョンアップはここまで
- 2023-02-12: Soft freeze 今はこの段階に入った状態
 - この期日までに testing に移行しているパッケージが次の安定版リリースに含まれる
 - この期日以降のパッケージの変更は小規模で的を絞った内容に限って行う
- 2023-03-12: Hard freeze
 - 重要なパッケージや autopkgtests がないパッケージは unblock リクエスト*9 が必要
- 未定: Full freeze
 - すべてのパッケージにおいて unblock リクエストが必要

追記:Debian 12 は 2023 年 6 月 10 日にリリースされました

2.11 Debian 12 の変更点

【ご注意ください】本発表のあった 2023 年 3 月現在の情報です。

詳細は各アーキテクチャの Debian 12 (bookworm) リリースノートをご確認ください。

apt に non-free-firmware セクションが追加

Debian Installer は non-free な firmware を含めた形で提供

```
- debian 11
deb http://deb.debian.org/debian/ bullseye main
- debian 12
deb http://deb.debian.org/debian/ bookworm main non-free-firmware
```

- Debian では「Debian 社会契約」でフリーと定めたパッケージのみインストーラに含めてきた
- 無線 LAN や GPU のドライバ等の動作には non-free なファームウェアが必要で、インストール時に ハードウェアが認識しないため不便という声がある
- Debian 開発者による投票で「Debian 社会契約」を変更して実現*10

/usr merge が適用された

- <https://wiki.debian.org/UsrMerge>

*8 https://release.debian.org/bookworm/freeze_policy.html

*9 debdiff を添えて unblock というバグ報告を行い、リリースチームのレビューを受ける必要あり

*10 https://www.debian.org/vote/2022/vote_003

- /bin、/sbin、/lib 系のディレクトリは /usr 配下へのシンボリックリンク

```
$ ls -l / | grep -E "bin|usr|lib"
lrwxrwxrwx 1 root root 7 3月 5 19:53 bin -> usr/bin
lrwxrwxrwx 1 root root 7 3月 5 19:53 lib -> usr/lib
lrwxrwxrwx 1 root root 9 3月 5 19:53 lib32 -> usr/lib32
lrwxrwxrwx 1 root root 9 3月 5 19:53 lib64 -> usr/lib64
lrwxrwxrwx 1 root root 10 3月 5 19:53 libx32 -> usr/libx32
lrwxrwxrwx 1 root root 8 3月 5 19:53 sbin -> usr/sbin
drwxr-xr-x 14 root root 4096 3月 5 19:53 usr
```

python 関連

- python3.11 を提供
- python2 系は完全に削除
 - Debian 11 で限定的に残っていた python2.7 は削除
 - python2 系でしか動かないスクリプトを使っている人は python3 に対応するよう直すこと
- pip install package で system-wide にインストールするときは `-break-system-packages` オプションが必要
 - at your own risk との説明が出てくる
 - PEP 668 の影響^{*11}
 - venv 環境を作る、pipx を使うなど作法が変わる

2.12 バグレポート

- 何かおかしい動作や不具合を見つけた場合はバグレポートをお願いします
- バグレポートの仕方 (レポートは英語で送る必要あり)
 - <https://www.debian.org/Bugs/Reporting.ja.html>
- バグレポートの前にちょっと相談してみたい方は、日本語の Debian JP メーリングリストや、SNS で相談してみてください
 - <https://www.debian.or.jp/community/ml/openml.html>
 - Twitter: @debian_jp

2.13 日本語による Debian の情報

- Debian JP Project
 - <https://www.debian.or.jp>
- 東京エリア Debian 勉強会
 - <https://tokyodebian-team.pages.debian.net/>
- 関西 Debian 勉強会
 - <https://wiki.debian.org/KansaiDebianMeeting>
- Twitter
 - @debian_jp

^{*11} <https://peps.python.org/pep-0668/>



3 Debian で PHP が動く Web サーバを構築してみる (apache2 / nginx)

dictoss さん

3.1 はじめに

最近 PHP で Web アプリの開発を少ししているため、Debian で PHP の Web アプリ開発をの開發する方法とサーバを構築する設定方法を調べてみました。なお、本記事における実行環境は Debian 12 bookworm とします。

3.2 Debian における PHP

3.2.1 PHP のインストール

Debian 12 bookworm では php というパッケージ名で PHP-8.2.4 を提供しています。

PHP を利用するには、以下の apt-get コマンドでインストールします。PHP は Web サーバ向けのスクリプト言語のため、依存関係の都合で apache2 が一緒にインストールされます。

```
# apt-get install php
(中略)
以下のパッケージが新たにインストールされます:
apache2 apache2-bin apache2-data apache2-utils libapache2-mod-php8.2 libapr1
libaprutil1 libaprutil1-dbd-sqlite3 libaprutil1-ldap libcurl4 liblua5.3-0
php php-common php8.2 php8.2-cli php8.2-common php8.2-opcache
php8.2-readline psmisc ssl-cert
アップグレード: 0 個、新規インストール: 20 個、削除: 0 個、保留: 0 個。
7,408 kB のアーカイブを取得する必要があります。
この操作後に追加で 31.2 MB のディスク容量が消費されます。
続行しますか? [Y/n]
```

インストールした php コマンドを実行してバージョンを確認してみます。

```
# php -v
PHP 8.2.4 (cli) (built: Mar 16 2023 14:24:40) (NTS)
Copyright (c) The PHP Group
Zend Engine v4.2.4, Copyright (c) Zend Technologies
with Zend OPcache v8.2.4, Copyright (c), by Zend Technologie
```

3.2.2 PHP のスクリプト開発 最初の一步

ファイルをまとめるディレクトリを作成

PHP のスクリプト開発を行うにあたり、まずはファイルを保存するディレクトリを作成します。

```
$ cd
$ mkdir phpapp
$ cd phpapp
```

phpinfo()

まずは PHP のスクリプト開発を始めるときのお約束である、phpinfo() を呼び出すファイルを作成します。

```
$ vi phpinfo.php
<?php
    phpinfo();
?>
```

php コマンドで phpinfo.php を実行すると、PHP の実行環境について詳細な情報が出力されます。

```
$ php phpinfo.php | head
phpinfo()
PHP Version => 8.2.4

System => Linux deb12-webapp-test 6.1.0-7-amd64 #1 SMP PREEMPT_DYNAMIC
Debian 6.1.20-1 (2023-03-19) x86_64
Build Date => Mar 16 2023 14:24:40
Build System => Linux
Server API => Command Line Interface
Virtual Directory Support => disabled
Configuration File (php.ini) Path => /etc/php/8.2/cli
Loaded Configuration File => /etc/php/8.2/cli/php.ini
(省略)
```

PHP の簡単な処理

PHP で for を使って、箇条書きの HTML を出力する printli.php を作成してみます。

```
$ vi printli.php
<!DOCTYPE html>
<html lang="ja">
<head>
<meta charset="UTF-8">
<title>test printli</title>
</head>
<body>
<ul>
<?php
for($i = 0; $i < 5; $i++){
    $s_link = sprintf("./detail/%d/", $i);
    $s_str = sprintf("article %d", $i);
    print "<li><a href=\"$s_link\">$s_str</li>\n";
}
?>
</ul>
</body>
</html>
```

Web ブラウザからディレクトリ直下にアクセスした場合に表示するトップページとして index.php を作成します。ページ内には、先ほど作成した phpinfo.php と printli.php へのリンクがある箇条書きの HTML が作成しておきます。

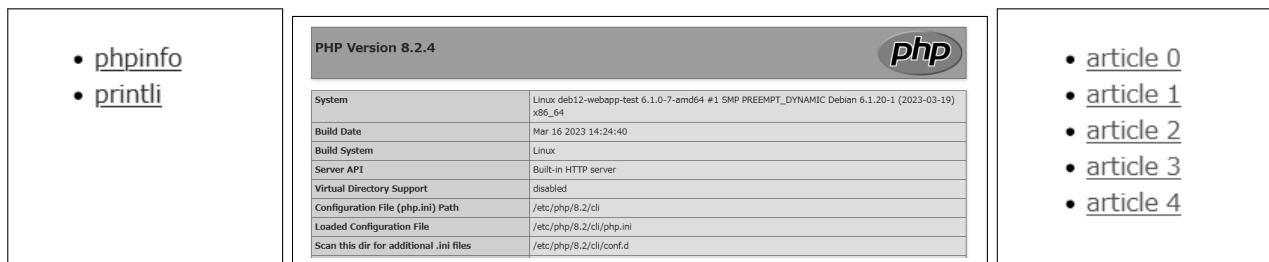
```
$ vi index.php
<!DOCTYPE html>
<html lang="ja">
<head>
<meta charset="UTF-8">
<title>test page</title>
</head>
<body>
<ul>
<li><a href="./phpinfo.php">phpinfo</li>
<li><a href="./printli.php">printli</li>
</ul>
</body>
</html>
```

PHP のビルトインウェブサーバーでの Web 画面表示

PHP には「ビルトインウェブサーバー」という機能があります*12。この機能を使うと PHP を動くように設定した Web サーバ を用意せずとも、ユーザのホームディレクトリ内にある PHP スクリプトを実行して Web 画面の表示と動作を確認することができます。

```
$ php -S 0.0.0.0:8000
[Mon Apr 10 22:56:08 2023] PHP 8.2.4 Development Server (http://0.0.0.0:8000) started
```

PHP を開発しているサーバの IP アドレスとポート番号を含めた "http://192.168.1.2:8000/" などの URL を Web ブラウザで開くと index.php の Web 画面が表示されます。Web ページには phpinfophp と printli.php へのリンクが表示されますので、クリックして画面表示を確認してみましょう。



PHP Version 8.2.4	
System	Linux deb12-webapp-test 6.1.0-7-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.20-1 (2023-03-19) x86_64
Build Date	Mar 16 2023 14:24:40
Build System	Linux
Server API	Built-in HTTP server
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/8.2/cli
Loaded Configuration File	/etc/php/8.2/cli/php.ini
Scan this dir for additional .ini files	/etc/php/8.2/cli/conf.d

- [phpinfo](#)
- [printli](#)

- [article 0](#)
- [article 1](#)
- [article 2](#)
- [article 3](#)
- [article 4](#)

図 1 index.php、phpinfo.php、printli.php

PHP パッケージマネージャ「composer」

PHP のパッケージマネージャに「composer」というコマンドがあります*13。composer を使うと PHP 向けのパッケージを配布する Web サイト*14 から、アプリの動作に必要な PHP のパッケージを一括でインストールやアップグレードすることができます。

composer は以下の apt-get コマンドでインストールできます。

```
# apt-get install composer
```

composer の使い方は、まずユーザ定義の設定ファイル "composer.json" を作成します。次に "composer install" コマンドを実行すると、composer.json の定義にしたがって PHP のパッケージをインストールします。コマンドの実行後は vendor ディレクトリ と composer.lock ファイルが生成され、インストールしたファイルと処理結果が保存されます。

```
$ vi composer.json
{
  "require": {
    "phpmailer/phpmailer": "~6.8.0"
  }
}
```

```
$ composer install
$ ls -l
composer.json composer.lock index.php phpinfophp printli.php vendor
合計 24
-rw-r--r-- 1 dictoss dictoss 67 4月 10 23:47 composer.json
-rw-r--r-- 1 dictoss dictoss 4039 4月 10 23:47 composer.lock
-rw-r--r-- 1 dictoss dictoss 216 4月 10 22:51 index.php
-rw-r--r-- 1 dictoss dictoss 24 4月 9 14:36 phpinfophp
-rw-r--r-- 1 dictoss dictoss 299 4月 10 23:25 printli.php
drwxr-xr-x 4 dictoss dictoss 4096 4月 10 23:47 vendor
```

*12 <https://www.php.net/manual/ja/features.commandline.websserver.php>

*13 <https://getcomposer.org/>

*14 規定では <https://packagist.org/> からダウンロードします。

3.2.3 PHP アプリケーションサーバ

mod_php

mod_php^{*15} とは apache で PHP を実行できるようにする組み込みモジュールであり、apache-1 時代から存在している PHP アプリケーションサーバの歴史ある実装です。Debian 12 bookworm では apache-2.4 向けに libapache2-mod-php8.2 パッケージで mod_php を提供しています。

linux 向けの apache-2.4 では MPM (マルチプロセッシングモジュール) が 3 つ含まれています (prefork、worker、event)。mod_php の実運用環境では apache を MPM マルチスレッドモードで使用することは推奨されていないため^{*16}、prefork MPM を選択する必要があります。

prefork MPM は 1 つの apache プロセスで同時に 1 つのクライアントのみを処理する方式であり、マルチスレッドの機能は実装されていないため動作がかなり安定しています。しかし、クライアント数が増えると PHP を処理する apache2 プロセスも連動して増えるためメモリの使用量がとても大きくなるデメリットがあります。

php-fpm

php-fpm^{*17} とは PHP における FastCGI^{*18} 実装です。なお、fpm とは "FastCGI Process Manager" の単語の頭文字をとった略称です。php-fpm には 高負荷のサイトで有用な機能が含まれています。Debian 12 bookworm では php-fpm というパッケージ名で php-fpm の機能を提供しています。

php-fpm は web サーバとは別の独立したプロセスで php を処理できるようバックグラウンドで待機し、web サーバから FastCGI プロトコルでの接続要求を受け入れて PHP スクリプトの実行結果を応答します。このとき、php-fpm のプロセスはクライアントの同時接続数に応じてプロセス数を自動で増減する仕組みがあります。

PHP の Web アプリケーションを提供するサーバでは多数のクライアントを処理するため、mod_php よりプロセス数が少なくメモリの利用効率が良い php-fpm を選択することを筆者はお勧めします^{*19}。

3.3 Debian で PHP が動く Web サーバの構築手順

3.3.1 apache2 + mod_php

apache2、php、libapache2-mod-php8.2 パッケージをインストールします。apache2 パッケージを単独でインストールすると MPM は event が有効になるのですが、mod_php である libapache2-mod-php8.2 パッケージのインストール処理で MPM が prefork へ切り替わります。

```
# apt-get install apache2 php libapache2-mod-php8.2
(中略)
Module mpm_event disabled.
Enabling module mpm_prefork.
apache2_switch_mpm Switch to prefork
apache2_invoke: Enable module php8.2
(省略)
```

有効になっている apache2 の MPM を確認すると prefork になっています。

^{*15} <https://www.php.net/manual/ja/install.unix.apache2.php>

^{*16} <https://www.php.net/manual/ja/install.unix.apache2.php>

^{*17} <https://www.php.net/manual/ja/install.fpm.php>

^{*18} 仕様は元々 fastcgi.com で公開していたが今は閉鎖されているため、コピーサイトの <https://fast-cgi.github.io/> で情報を確認できます。

^{*19} php-fpm で PHP を実行する場合は、apache は event MPM を利用することができます。event MPM は linux では epoll を使ったイベント駆動な実装になっており、少ないプロセス数とスレッド数で多数のクライアントを処理することができます。

```
# cd /etc/apache2/mods-enabled
# ls -l mpm*
lrwxrwxrwx 1 root root 34  4月  9 14:34 mpm_prefork.conf -> ../mods-available/mpm_prefork.conf
lrwxrwxrwx 1 root root 34  4月  9 14:34 mpm_prefork.load -> ../mods-available/mpm_prefork.load
```

もし、MPM が prefork になっていない場合は、a2dismod で mpm_event を無効化し、a2enmod で mpm_prefork を有効化してください。

```
# a2dismod mpm_event
Module mpm_event disabled.
To activate the new configuration, you need to run:
systemctl restart apache2

# a2enmod mpm_prefork
Considering conflict mpm_event for mpm_prefork:
Considering conflict mpm_worker for mpm_prefork:
Enabling module mpm_prefork.
To activate the new configuration, you need to run:
systemctl restart apache2
```

apache2 を再起動します。

```
# systemctl restart apache2
```

Web ブラウザで phpinfo を表示する画面を開きます。一番上部にある表の ”Server API” 行の設定値が ”Apache 2.0 Handler” になっていれば mod_php で動作しています。

PHP Version 8.2.4		
System	Linux deb12-webapp-test 6.1.0-7-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.20-1 (2023-03-19) x86_64	
Build Date	Mar 16 2023 14:24:40	
Build System	Linux	
Server API	Apache 2.0 Handler	
Virtual Directory Support	disabled	
Configuration File (php.ini) Path	/etc/php/8.2/apache2	
Loaded Configuration File	/etc/php/8.2/apache2/php.ini	
Scan this dir for additional .ini files	/etc/php/8.2/apache2/conf.d	

図 2 phpinfo 画面 (apache2 + mod_php)

3.3.2 apache2 + php-fpm

apache2、php、php-fpm パッケージをインストールします。php パッケージの依存関係の都合で mod_php である libapache2-mod-php8.2 がインストールされてしまいます。


```
# apt-get install apache2 php php-fpm

(中略)
下のパッケージが新たにインストールされます:
apache2 apache2-bin apache2-data apache2-utils libapache2-mod-php8.2
libapr1 libaprutil1 libaprutil1-dbd-sqlite3 libaprutil1-ldap libcurl4
liblua5.3-0 php php-common php-fpm php8.2 php8.2-cli php8.2-common
php8.2-fpm php8.2-opcache php8.2-readline psmisc ssl-cert

(中略)
php8.2-fpm (8.2.4-1) のトリガを処理しています ...
NOTICE: Not enabling PHP 8.2 FPM by default.
NOTICE: To enable PHP 8.2 FPM in Apache2 do:
NOTICE: a2enmod proxy_fcgi setenvif
NOTICE: a2enconf php8.2-fpm
NOTICE: You are seeing this message because you have apache2 package installed.
libapache2-mod-php8.2 (8.2.4-1) のトリガを処理しています ...
```

php-fpm で動作させたい場合は mod_php をインストールしておくことと設定のコンフリクトを起こすため libapache2-mod-php8.2 パッケージを削除しておきます。

```
# apt-get purge libapache2-mod-php8.2
```

パッケージのインストール時に表示された NOTICE に従い、apache2 のモジュールと設定を有効化します。

```
# a2enmod proxy_fcgi setenvif
Considering dependency proxy for proxy_fcgi:
Enabling module proxy.
Enabling module proxy_fcgi.
Module setenvif already enabled
To activate the new configuration, you need to run:
    systemctl restart apache2

# a2enconf php8.2-fpm
Enabling conf php8.2-fpm.
To activate the new configuration, you need to run:
    systemctl reload apache2
```

apache2 の MPM を prefork から event へ切り替えます。

```
# a2dismod mpm_prefork
Module mpm_prefork disabled.
To activate the new configuration, you need to run:
    systemctl restart apache2

# a2enmod mpm_event
Considering conflict mpm_worker for mpm_event:
Considering conflict mpm_prefork for mpm_event:
Enabling module mpm_event.
To activate the new configuration, you need to run:
    systemctl restart apache2
```

php8.2-fpm、apache2 を再起動します。

```
# systemctl restart php8.2-fpm
# systemctl restart apache2
```

Web ブラウザで phpinfo を表示する画面を開きます。一番上部にある表の "Server API" 行の設定値が "FPM/FastCGI" になっていれば php-fpm で動作しています。

3.3.3 nginx + php-fpm

nginx、php-fpm パッケージをインストールします。

```
# apt-get install nginx php-fpm

(中略)
下のパッケージが新たにインストールされます:
nginx nginx-common php-common php-fpm php8.2-cli php8.2-common php8.2-fpm php8.2-opcache php8.2-readline
```

php8.2-fpm、nginx を再起動します。

System	Linux deb12-webapp-test 6.1.0-7-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.20-1 (2023-03-19) x86_64
Build Date	Mar 16 2023 14:24:40
Build System	Linux
Server API	FPM/FastCGI
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/8.2/fpm
Loaded Configuration File	/etc/php/8.2/fpm/php.ini
Scan this dir for additional .ini files	/etc/php/8.2/fpm/conf.d

図 3 phpinfo 画面 (apache2 + php-fpm)

```
# systemctl restart php8.2-fpm
# systemctl restart nginx
```

Web ブラウザで phpinfo を表示する画面を開きます。なぜか phpinfo.php ファイルがサーバで実行されず、ダウンロードされてしまいました。

nginx の設定を確認すると、`/etc/nginx/sites-available/default` の `fastcgi` の中継先となる UNIX ドメインソケットのパスがコメントで無効になっていました。

```
# grep -nr fastcgi_pass /etc/nginx
/etc/nginx/sites-available/default:60: #       fastcgi_pass unix:/run/php/php7.4-fpm.sock;
/etc/nginx/sites-available/default:62: #       fastcgi_pass 127.0.0.1:9000;
```

`/etc/nginx/sites-available/default` を変更します。

```
# vi /etc/nginx/sites-available/default
- #location ~ /\.php$ {
- #       include snippets/fastcgi-php.conf;
- #
- #       # With php-fpm (or other unix sockets):
- #       fastcgi_pass unix:/run/php/php7.4-fpm.sock;
- #       # With php-cgi (or other tcp sockets):
- #       fastcgi_pass 127.0.0.1:9000;
- #}
+ location ~ /\.php$ {
+       include snippets/fastcgi-php.conf;
+
+       # With php-fpm (or other unix sockets):
+       fastcgi_pass unix:/run/php/php8.2-fpm.sock;
+       # With php-cgi (or other tcp sockets):
+       #fastcgi_pass 127.0.0.1:9000;
+ }
```

変更した設定を反映するため、nginx を再起動します。

```
# systemctl restart nginx
```

Web ブラウザで phpinfo を表示する画面を開きます。一番上部にある表の”Server API” 行の設定値が”FPM/FastCGI” になっていれば php-fpm で動作しています。

System	Linux deb12-webapp-test 6.1.0-7-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.20-1 (2023-03-19) x86_64
Build Date	Mar 16 2023 14:24:40
Build System	Linux
Server API	FPM/FastCGI
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/8.2/fpm
Loaded Configuration File	/etc/php/8.2/fpm/php.ini
Scan this dir for additional .ini files	/etc/php/8.2/fpm/conf.d

図 4 phpinfo 画面 (nginx + php-fpm)

3.4 終わりに

Debian 12 bookworm で PHP を実行できる Web サーバを構築する設定方法を説明しました。apache2 のモジュールや設定を有効化・無効化する `a2enmod`、`a2dismod`、`a2enconf`、`a2disconf` コマンドや、nginx の php-fpm の中継先の設定を正しく行わないと PHP をうまく実行できないため、注意が必要です。

そのほか、apache2 の MPM 設定、nginx の worker 設定、php-fpm のプロセス数設定をチューニングするとサーバの応答性能をさらに引き出すことができるため、こだわりがある人は挑戦してみてください。

4 DDTP 及び DDTSS の紹介

dictoss さん



4.1 DDTP とは

- The Debian Description Translation Project
- Debian パッケージの説明文の翻訳を取り組むプロジェクト
- <https://www.debian.org/international/l10n/ddtp.ja.html>
- 翻訳状況の統計情報、翻訳作業の支援ツールを提供
- Debian パッケージの翻訳を行うと以下の場所などに翻訳文が表示される
 - <https://packages.debian.org/ja/>
 - `apt show packagename`
- パッケージ説明文翻訳の言語別の進捗
 - <https://ddtp.debian.org/stats/stats-sid.html>
 - Required、Important、Standard、Optional、Extra の区分は Debian パッケージの優先度^{*20}
 - * Required、Important、Standard のパッケージは利用者が多いため優先的に翻訳するとよい
 - Popcon500、PopconRank は popularity-contest^{*21} パッケージで収集されたパッケージの利用状況のランキング
 - * Popcon500 のパッケージは利用者が多いため優先的に翻訳するとよい
 - 表のセルにマウスを乗せて少し待つとツールチップが表示され、未翻訳のパッケージ名が表示される

4.2 DDTSS とは

- Debian パッケージの説明文の翻訳とレビューを行う Web サイト
 - Debian Distributed Translation Server Satellite
 - <https://ddtp.debian.org/ddtss/index.cgi/xx>
 - sid のパッケージが作業対象
 - 翻訳を行い、3 人のレビューが OK になると反映される
 - アカウントの登録が必要

^{*20} <https://www.debian.org/doc/manuals/debian-faq/pkg-basics.ja.html#priority>

^{*21} <https://popcon.debian.org/>

4.3 そのほかの翻訳ツール

- <https://hosted.weblate.org/>
 - プロジェクト横断でドキュメントの翻訳作業を支援する Web サイト
- Debian では以下のドキュメントの翻訳作業が行える
 - <https://hosted.weblate.org/projects/debian-installer/>
 - <https://hosted.weblate.org/projects/debian-installation-guide/>
 - <https://hosted.weblate.org/projects/debian-reference/>
 - <https://hosted.weblate.org/projects/debian-handbook/>
 - <https://hosted.weblate.org/projects/debexpo/>

4.4 リリースノート

- 基本的な情報 <https://wiki.debian.org/ReleaseNotes>
- 翻訳ファイル置き場
 - <https://salsa.debian.org/ddp-team/release-notes/>
 - 翻訳 web サイトはないため、po ファイルを修正してマージリクエストを出します
- 翻訳状況
 - Debian 11 bullseye 版 <https://www.debian.org/releases/bullseye/statistics.html>
 - Debian testing 版 <https://www.debian.org/releases/testing/statistics.html>

4.5 まとめ

- DDTP の取り組みを紹介しました
- DDTSS の Web サイト <https://ddtp.debian.org/ddtss/index.cgi/xx>
- できるだけ翻訳作業を進めましょう

5 Debian パッケージング Open Build Service 編

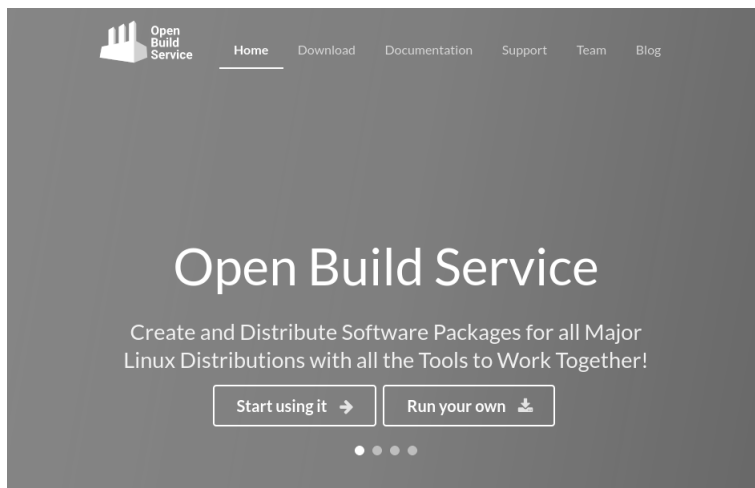
林 さん

本記事は Debian 勉強会 (2023 年 4 月開催) にて公開した資料を冊子向けに編纂・収録したものです。

資料については、Rabbit Slide Show^{*22}にて公開されています。

今回は Open Build Service を試す機会があったのでその紹介をします。

5.1 Open Build Service とは



公式サイトは <https://openbuildservice.org/> で、openSUSE の開発に利用されているソフトウェアです。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であり、OBS のインスタンスが <https://build.opensuse.org/> として運用されています。

^{*22} <https://slide.rabbit-shocker.org/authors/kenhys/tokyodebian-obs-howto-202304/>



Open Build Service について知るには、「第 1 回 Open Build Service 道場」^{*23}が参考になります。2012 年の資料ではあるものの参考になるはずでず。他にも、2014 年の資料ではありますが、「Open Build Service 道場」^{*24}パッケージの新規作成編 作りかけバージョンというのもあります。

Open Build Service では、deb や rpm のビルドができる (Debian, Ubuntu, Fedora, openSUSE) ことに加え、ビルドしたパッケージのリポジトリを公開できます。パッケージおよび、そのリポジトリを作成できるという意味では、類似サービスがいくつかあります。

Packagecloud では <https://packagecloud.io> リポジトリのホスティングを提供しています。ただし、パッケージのビルドは自前で行う必要があります。

Launchpad^{*25}もよく知られたサービスです。リポジトリのホスティングを提供し、パッケージのビルドもおまかせにできます。しかし、サポートされている Ubuntu のリリースのみです。

Open Build Service の便利なところはどこでしょうか。1 つには、RPM 系だけでなく、Deb もサポートしているところです。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ますし、パッケージの署名用の鍵のメンテナンス不要です。

また、ミラーを利用できる^{*26}のもポイントです。<https://mirrorcache.opensuse.org> を指定してあれば、適宜適切なミラー^{*27}が自動的に選択されます。また、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ます。

他にも次のような利点があります。

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

しかし、いいことばかりではありません。次の場合のように、あまりマッチしないこともあるかもしれません。

- <https://mirrorcache-jp.opensuse.org> へのミラーはそれほど早くないので、1 日くらいみとくとよい。(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)
- 独自プロジェクトにするには手間と時間がかかる
- 未対応のディストリビューションもある (例: AmazonLinux)

^{*23} <https://www.slideshare.net/ftake/1-open-build-service>

^{*24} <https://www.slideshare.net/ftake/obs-dojoadv>

^{*25} <https://launchpad.net>

^{*26} <https://en.opensuse.org/MirrorCache>

^{*27} 日本の場合は <https://mirrorcache-jp.opensuse.org> です。

5.2 OBS を利用する方法

OBS を利用するための初期セットアップ方法*28は次のとおりです。

- <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ます。
- osc コマンドを利用できるようにする (例: `sudo apt install -y osc` でインストールできます。)

必要なソフトウェアがインストールできたら、プロジェクトをチェックアウトしてみましょう。osc checkout home:(ユーザー名) でプロジェクトをチェックアウトできます。(SVNっぽい感じというと伝わる人もいられるかもしれませんが。) 初回のチェックアウトの際アカウント情報を入力することで ~/.config/osc/osrcrc が作られます。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ので初見ではびっくりするかもしれません。

次に、プロジェクトの設定を行います。メタ情報の指定が必要です。プロジェクトのメタ情報とは、どのディストリビューションとひもづけるかというようなことを意味します。実際にひもづけるには、meta サブコマンドを利用して、メタ情報を記述したファイルを指定します。(ファイルを指定しない場合、エディタが起動して入力する)

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

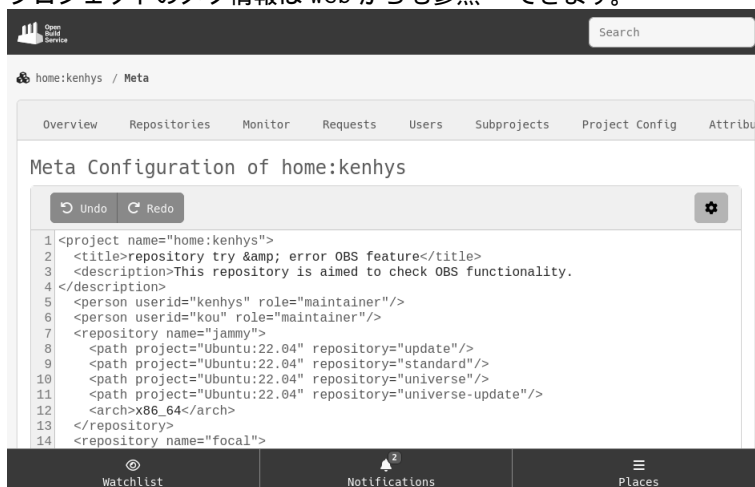
メタ情報を記述したファイルの内容は次のようなものです。ビルド対象のディストリビューションや有効にするリポジトリを指定します。

```
<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には、role をプロジェクトのメタ情報に含めると実現できます。

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

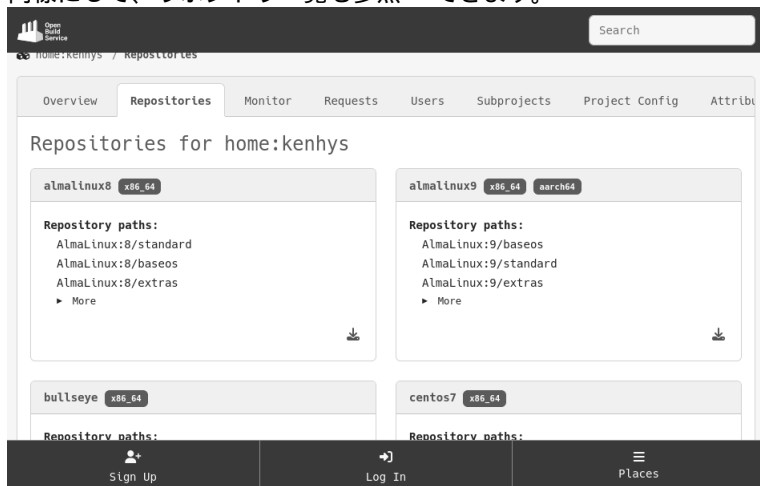
プロジェクトのメタ情報は Web から参照*29できます。



*28 https://en.opensuse.org/openSUSE:Build_Service_Tutorial

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

同様に、リポジトリ一覧も参照*30できます。



ちなみに、OBS が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どうかは、`osc dists` コマンドを実行して確認できます。

```
$ osc dists
```

なお、特定のディストリビューションでは、明示的にフラグを設定しないとパッケージがうまくビルドできないということがあります。

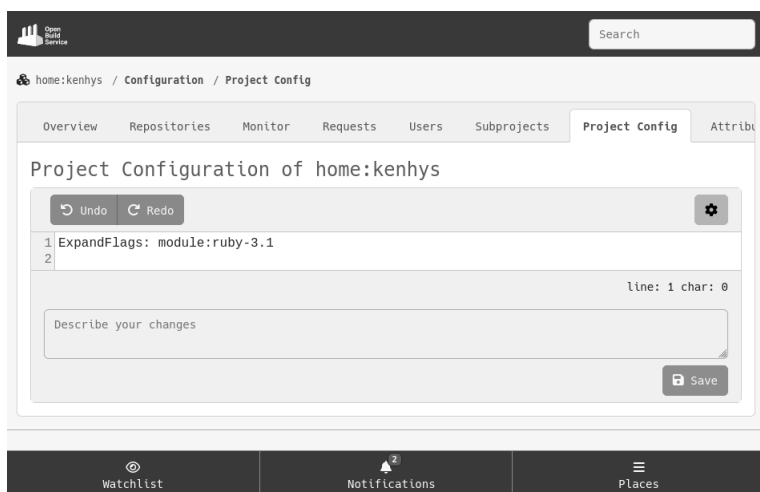
例えば、AlmaLinux:8 などのように AppStream でモジュールとして各種パッケージが提供されている場合 (`dnf module list` で対象を確認できる)、OBS では既定でモジュールは有効になっていません。つまり、AppStream かつモジュールのパッケージを `spec` の `BuildRequires:` に指定すると `unresolvable` となってビルドすらされないということを意味します。ログには `nothing provides: xxxx` とそっけないメッセージがでるだけなのでわかりにくいと言えます。

このような場合に特定のモジュールを有効にするには、プロジェクトにフラグを設定する必要があります。

```
$ osc meta prjconf home:kenhys --file prjconf.txt
```

設定するフラグの内容は例えば次のようなものです。

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



プロジェクトにどのようなフラグを設定すべきかについては、ドキュメント*31を参照してください。

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

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

こちらも同様に Web から設定を参照^{*32}できます。

5.3 パッケージごとにすること

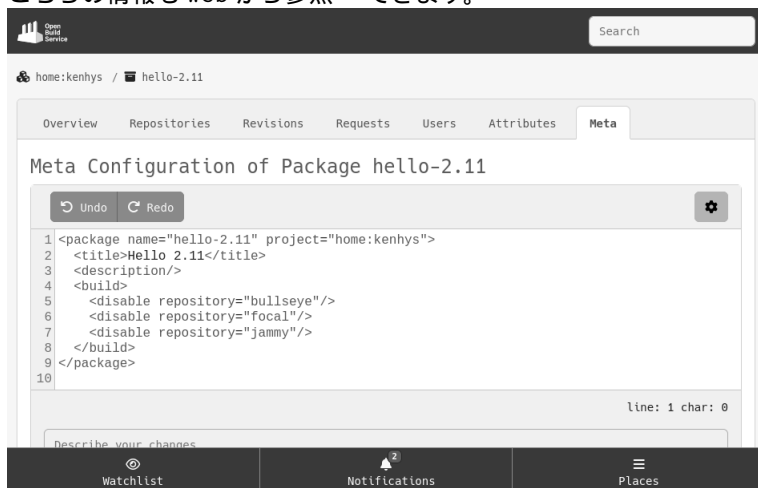
プロジェクトの設定ができたなら、次にパッケージのメタ情報を設定する必要があります。特定のディストリビューションに対応していない場合、パッケージごとと無効にします。プロジェクトのメタデータで設定したリポジトリ名を指定したり、<build>のターゲットとして<disable>を指定します。

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

パッケージのメタ情報は例えば次のようなものです。(例: プロジェクトでは Debian 系のビルドを有効にしたが、hello パッケージではビルドしないように無効化する場合)

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

こちらの情報も Web から参照^{*33}できます。



5.4 OBS でパッケージをビルドする方法

ここまでで、パッケージに関するメタ情報まで設定できました。それでは、実際にパッケージをビルドする方法を紹介していきます。

.rpm のビルドするには 2 つやりかたがあります。1 つは、SRPM から作成する方法です。例えば、既存の別サービスから移行するなら、SRPM をインポートするのがお手軽です。もう 1 つは、.spec と関連するソースアーカイブから作成するやりかたです。

SRPM を使う場合、home:(ユーザー名) で次のコマンドを実行します。

```
$ osc importsrpkp -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"
```

このようにすることでパッケージをビルドできます。

.spec や各種ソースアーカイブを使う場合には、次のように osc meta でパッケージに関するメタ情報を追加し、アーカイブを追加します。

^{*32} [urlhttps://build.opensuse.org/projects/home:\(ユーザー名\)/prjconf](https://build.opensuse.org/projects/home:(ユーザー名)/prjconf)

^{*33} [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)

```
$ 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"
```

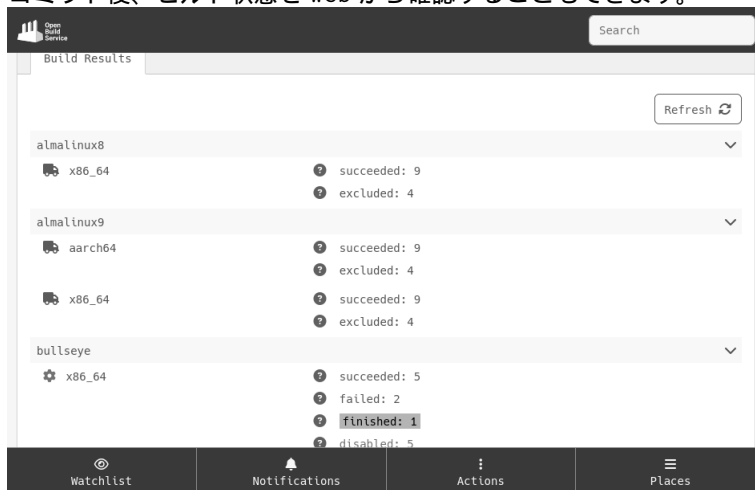
ファイルを経由して指定するパッケージのメタ情報は、次のようなものです。

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

.deb のビルドするには、次のようにします。 .dsc, orig.tar.gz, debian.tar.xz をあらかじめ追加してからコミットします。

```
$ 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"
```

コミット後、ビルド状態を Web から確認することもできます。

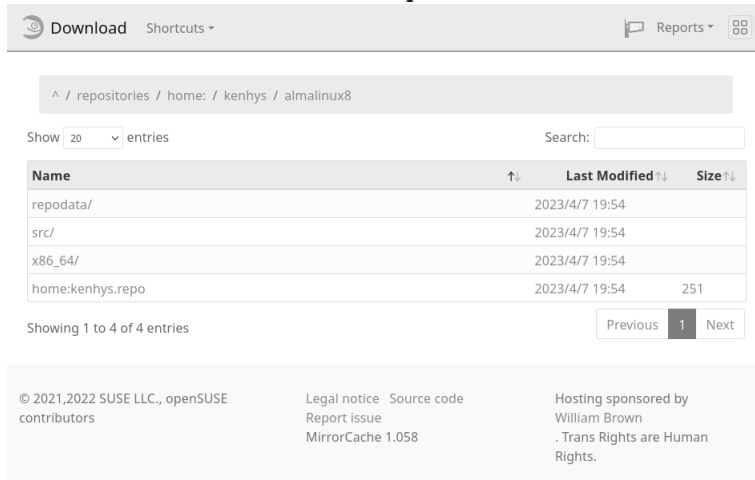


あるいは、osc results でも確認できます。

5.5 リポジトリの利用方法

ビルドしたパッケージを利用するには次のようにします。

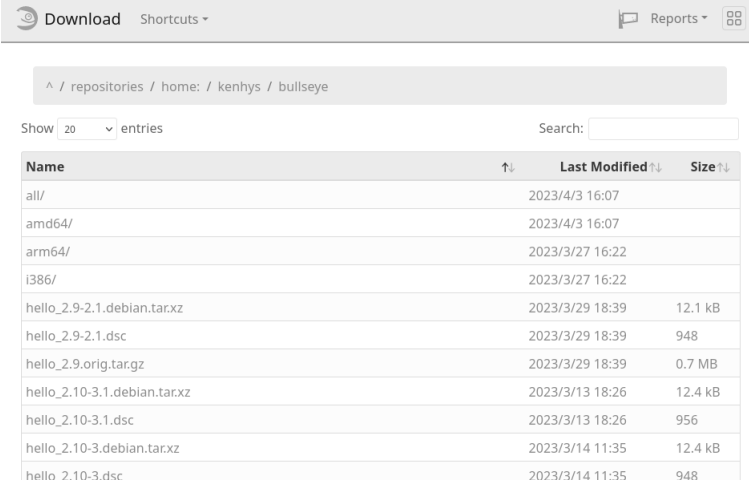
例えば、AlmaLinux:8 であれば.repo が公開されているのでそれを利用できます。



Name	Last Modified	Size
repodata/	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

```
$ 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
```

Debian 系の場合は、Release.key が公開されているのでそれを利用できます。



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、keyring に変換して使うとよいでしょう。

```
$ 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/
```

keyring に変換できたら、あとは apt-line を追加します。

```
# 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) /
```

5.6 OBS でありがちなこと

OBS を使っていると、いくつかはまるポイントがあります。OBS でパッケージのビルドがはじまらないときは、後述する問題がないか確認しましょう。

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か、不足しているプロジェクトのフラグを追加することで解決できます。

Docker ではビルドできるのに OBS ではビルドできない、ということもあります。root ユーザーでのビルドを前提していると失敗するので、一般ユーザーでビルドできるように修正すると解決します。

複数バージョンを公開したいがビルドされなくなった場合は、バージョンつけずに複数ソースをコミットしたときに発生します。バージョンをつけてパッケージを管理することで解決できます。(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)

また、home じゃないプロジェクトの作成難易度が高かったりもします。^{*34}

^{*34} <https://build.opensuse.org/project/new> で新規プロジェクトを作成できると案内されていますが、home: 配下しか作成できません。science: とか server:mail などのカテゴリがありますが、あくまでこちらは openSUSE の開発で使われるプロジェクト向けなので、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とされています。しかし、<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ますが、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さそうです。これは、既知の問題^{*35}ですが修正されていません。<admin@opensuse.org> に連絡すると <https://progress.opensuse.org/> に issue を作ってもらえますが opensuse-admin-obs にアサインしてもらえないと気づかれません。OBS とはチームが別なので気づいてもらえるの待ちということになります。

5.7 OBS はどんな用途に向くのか？

ここまでいろいろ紹介してきましたが、OBS の利用がおすすめできるのは次のような場合です。

- 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 かつ openSUSE 専用にメンテナンスするのでなければ、結局のところ home: 配下にプロジェクトを作成するのが（野良パッケージ感があるものの）楽かもしれません。

6 Debian パッケージング アップロード編

林 さん



本記事は Debian 勉強会（2023 年 1 月開催）にて公開した資料を冊子向けに編纂・収録したものです。

資料については、Rabbit Slide Show^{*36}にて公開されています。

6.1 Debian パッケージのアップロードの機会

パッケージのメンテナンスをする人は主に次の 3 つのカテゴリに分けられます。

- コントリビューターによるアップロード（例： mentors.d.n へのアップロードなどをする）
- DM/DD: 既存のパッケージのメンテナンスをする
- DD: 新規パッケージのスポンサー（例： ftp-master へのスポンサーアップロードなど）

どのカテゴリであっても、パッケージのメンテナンスをする際にはアップロードツールを普通は使います。というわけで今回はパッケージのアップロードツールのおすすめについて紹介します。

6.2 アップロードに用いられるツールの比較

アップロードに用いられるツールは概ね以下の 3 つがあります。^{*37} ^{*38}

- dput（圧倒的に利用者が多い）
- dupload（dput-ng より使われている）
- dput-ng（一部の人が使っている）

それぞれどれくらい使われているのか popcon のデータをもとに示します。

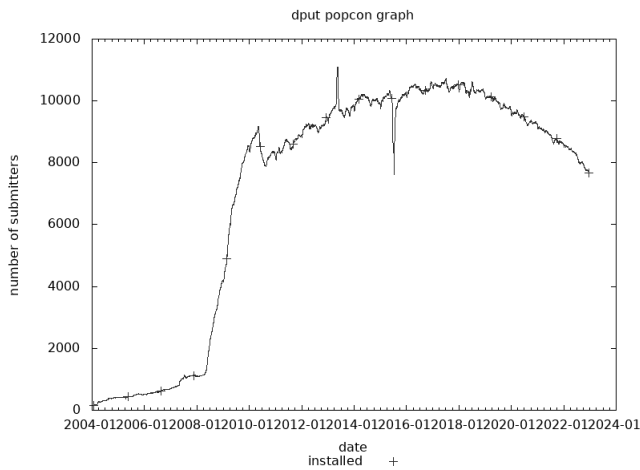
dput は最も使われている^{*39}ツールです。インストール数に若干減少傾向がみられますが、それでも他の追従を許しません。

^{*36} <https://slide.rabbit-shocker.org/authors/kenhys/tokyodebian-dputng-howto-202301/>

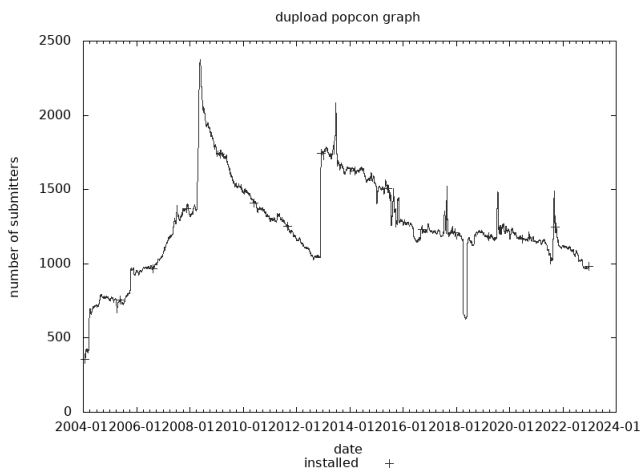
^{*37} <https://www.debian.org/doc/manuals/developers-reference/tools.ja.html#package-uploaders>

^{*38} <https://www.debian.org/doc/manuals/maint-guide/upload.ja.html>

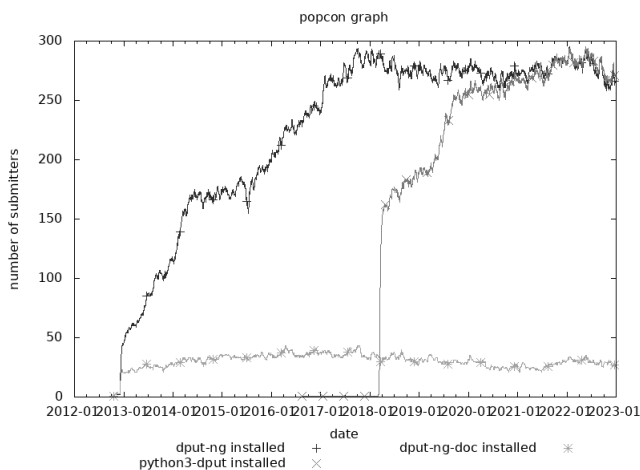
^{*39} <https://qa.debian.org/popcon.php?package=dput>



dupload はどちらかというトリポジトリのメンテナンスのために裏方として動作していることの多いツールです。dput ほどではありませんがよく使われているツールである^{*40}といえるでしょう。



dput-ng は dput や dupload に比べると後発のツールです。利用者を徐々に増やしつつあります。^{*41}



では、それぞれのツールの特徴をおさらいしておきましょう。dput には次のような特徴があります。

- ほぼデファクトスタンダードのアップロードツール
 - devscripts の recommends でインストールされる

^{*40} <https://qa.debian.org/popcon.php?package=dupload>

^{*41} <https://qa.debian.org/popcon.php?package=dput-ng>

- packaging-dev の依存でインストールされる
- ~/.dput.cf でユーザーごとにカスタマイズ
 - PPA にアップロードするときとかカスタマイズが必要なことがある

一方後発の dput-ng^{*42}には次のような特徴があります。

- dput との設定ファイルの互換性を維持しつつも方向性が異なるプロジェクト
- フックの仕組みでチェック機能を拡張可能 (例: 3rd パーティーが開発したチェッカーを利用できる)
- 独自書式の設定ファイルで挙動を柔軟に変えることもできる
- DM のパーミッションの操作もできる (dcut の機能強化)
- dput と併用はできない
 - dput-ng をインストールすると dput は削除される
 - ドキュメントは dput よく知っている前提

どちらかという dput すでによくわかっている人向けの側面があります。利用可能なフックの説明はなく、実装についてのページがある^{*43}程度だったりするので、dput(1) もしくは dput(5) の参照をおすすめします。

dput-ng でサポートされているフックについては、dirt を使って調べることができます。dirt list コマンドを実行してみてください。次のような結果が得られるはずです。

```
% dirt list
WARNING: This command is not completed yet. Interface and behavior changes are expected
in future releases

pre-upload hooks:
  required-fields: check whether a field is present and non-empty in the changes file
  checksum: verify checksums before uploading
  protected-distribution: warn before uploading to distributions where a special policy
applies
  lintian: runs lintian before the upload
  gpg: check GnuPG signatures before the upload
  allowed-distribution: check whether a local profile permits uploads to the target
distribution
  suite-mismatch: check the target distribution for common errors
  supported-distribution: check whether the target distribution is currently supported
(using distro-info)
  check-debs: makes sure the upload contains a binary package

post-upload hooks:
  impatient: Tell me when dinstall will pick this up
```

dupload には次のような特徴があります。

- dput, dput-ng と同様にアップロードするためのもの
- dput, dput-ng より機能が少ない
- 設定 (.dupload.conf) に従ってアナウンスメールを送信する機能あり (.upload があればスキップしたり、チェックサムの確認機能ぐらい)

dupload はどちらかと debian アーカイブを運用していてそのために使うという性質が強いので、パッケージのメンテナンスをしている開発者の観点だと dput と dput-ng のどちらかを使うのがよさそうということになるでしょう。

6.3 チェック機能からみた各ツールの違いを比較する

各ツールのチェック可能な項目は次のとおりです。

- dput で利用可能なチェック: 7
- dput-ng で利用可能なチェック: 10

^{*42} <https://dput.readthedocs.io/en/latest/>

^{*43} <https://dput.readthedocs.io/en/latest/library/hooks/index.html>

- dupload で利用可能なチェック: (省略)

まずは dput で利用可能なチェック項目をみていきましょう。

- Unique upload (アップロードしたときのログがあればアップロードを中断する)
- Required package files (必要なファイルが含まれていないときにアップロードを中断する)
- Successful local install (パッケージをインストールした状態でないときにアップロードを中断する。既定では無効。)
- Cryptographic signature (GPG 署名していないときにアップロードを中断する。既定で有効。)
- Package files content match (サイズやハッシュ値が一致していないときにアップロードを中断する。既定で有効。)
- Distribution name is permitted for the remote host (アップロード先が許容しない distribution の場合にアップロードを中断する。サーバー側設定に依存)
- Lintian success (lintian が失敗したときにアップロードを中断する。既定で無効)

次に、dput-ng で利用可能なチェック項目をみていきましょう。

- allowed-distribution (プロファイルに応じてアップロードしてもよいかチェックする。例: debian/changelog で UNRELEASED のままアップロードしようとしたら止める)
- check-debs フック (ソース or バイナリどちらが含まれるべきかチェックする。例: PPA にバイナリパッケージをアップロードしようとするとエラー)
- checksum フック (アップロード前のパッケージのチェックサムがあっているかをチェックする)
- gpg フック (アップロード前に GnuPG で署名してあるかをチェックする。署名していなかったら署名をうながし、アップロードできる)
- impatient フック (次に dinstall が実行されるまでの時間を通知する)
- lintian フック (アップロード前に lintian を実行する)
- protected-distribution フック (特別なポリシーが適用される場所へのアップロード前に警告する。例: testing-proposed-updates, proposed-updates, oldstable, stable, testing, stable-security, oldstable-security, testing-security などにアップロードしようとしたときに確認が求められる)
- required-fields フック (指定された項目があるかをチェックする。例: .changes ファイルにプロファイルで定義した required-fields に指定されたものが含まれていないと警告する)
- suite-mismatch フック (changelog と .changes のディスクリプションが一致しない場合、警告する。例: unstable で experimental が混在しているとき)
- supported-distribution フック (.changes の Distribution がサポートされていないときに警告する。例: サポートの有無は python3-distro-info を使って /usr/share/distro-info/debian.csv を参照する)

個人的には dput-ng がサポートしている gpg フックが (署名忘れていたときに) 便利なのでおすすめです。

ほかにも、dput-ng にはサンプルとして利用できるフックが付属しています。

- bd-blacklist (Build-Depends に対象が含まれていたら失敗する)
- clojure-arno-tester (メンテナーが arno@debian.org だったら失敗する)
- twitter (アップロード後に tweet する)

サンプルを参考にいろいろ拡張してみるのも面白いかもしれません。

6.4 各アップロードツールの機能比較

あらためて、各ツールの機能を比較してみました。

表 1 アップロードツールの機能比較

	dupload	dput	dput-ng
提供コマンド	dupload	dput,dcut	dput,dcut
設定ファイル	~/dupload.conf	~/dput.cf or ~/.config/dput/dput.cf	~/dput.cf or 独自
フックの追加実装	非対応	非対応	対応
対応プロトコル	ftp,http,https,scp,scp,rsync,copy	ftp,http,https,scp,rsync,local	ftp,http,https,scp,local
メール連携	対応	非対応	非対応
DM の権限操作	非対応	非対応	dcut 追加機能

拡張性や、対応プロトコルといったところに違いがあることがわかります。

```
% dcut dm \  
  --uid 66DEF15282990C2199EFA801A8A128A8AB1CEE49 \  
  --allow zarchive  
Uploading commands file to  
ftp.upload.debian.org (incoming: /pub/UploadQueue/)  
Picking DM Andrea Pappacoda  
(Tachi's main key) <andrea@pappacoda.it>  
with fingerprint 66DEF15282990C2199EFA801A8A128A8AB1CEE49  
Uploading kenhys-1673679146.dak-commands to ftp-master
```

dput-ng 版では dcut で DM の許可を変更できる、というのは上記の操作で権限を与えられるということです。

ここまでで、各アップロードツールの概要を比較してみました。dput-ng には dput にはないプロファイルという考え方があります。次はそのプロファイルについて説明します。

6.5 dput-ng でのプロファイルの考え方

dput-ng は基本的に dput と同じ設定ファイルを読み込みますが、独自形式の設定ファイルもサポートしています。それが、dput と互換性のないフォーマット (JSON) です。

独自形式では、メタ情報を拡張することで各種需要にあわせてプロファイル (例: debian というメタ情報から mentors-ftp プロファイルを派生) を定義できるようになっています。基本的な設定をメタ情報として定義し、サイトごとの設定をプロファイルで指定して上書きするということができます。

dput の場合、https://mentors.debian.net/intro-maintainers/ に FTP 利用者向けに ~/.dput.cf に設定する内容がサンプルとして提示されています。

```
[mentors-ftp]  
fqdn = mentors.debian.net  
login = anonymous  
progress_indicator = 2  
passive_ftp = 1  
incoming = /pub/UploadQueue/  
method = ftp  
allow_unsigned_uploads = 0  
# Allow uploads for UNRELEASED packages  
allowed_distributions = .*
```

これと同等の設定は dput-ng ではあらかじめ /etc/dput.d/profiles/mentors.json で定義されています。

```

{
  "-hooks": [
    "check-debs",
    "allowed-distribution"
  ],
  "allowed_distributions": ".*",
  "fqdn": "mentors.debian.net",
  "incoming": "/pub/UploadQueue/",
  "login": "anonymous",
  "meta": "debian",
  "method": "ftp"
}

```

継承元の/etc/dput.d/metasp/debian.json 定義は次のとおりです。

```

{
  "allow_dcut": true,
  "allowed-distribution": {
    "codename-groups": [
      "general", "backport", "rm-managed"
    ]
  },
  "codenames": "debian",
  "hooks": [
    "allowed-distribution",
    "protected-distribution",
    "checksum",
    "suite-mismatch",
    "gpg"
  ],
  ...(省略)...
}

```

プロファイルの継承により、差分を定義すればよくなっていることがわかります。

6.6 結局、アップロードツールはどれを使うのがよいのか

dupload がおすすめなのは次のような場合です。

- メール連携が使いやすい (リポジトリを運用しているのでアップロード時に通知をしたい)
- シンプルな機能で十分派
- チェック機能は少なめでもよい
- rsync でアップロード必須

dput がおすすめなのは次のような場合です。

- メジャーなものを使いたい派
- チェックは dput のもので十分派
- rsync でアップロード必須

dput-ng がおすすめなのは、次のような場合です。

- チェック機能を強化したい派
- アップロード時に署名もまとめて実行したい
- DM の権限変更もしたい

dput と dput-ng、設定ファイルの互換性あるものの機能性に違いがあります。ご自身の実現したいことができるかで、よりよいツールを選択してみてください。個人的にはアップロード時に署名もできるので dput-ng がオススメです。

7 Analysis of the Debian Developer Survey (2022)

発表者: dictoss さん



Debian Developer へのアンケート Analysis of the Debian Developer Survey (2022) を読んでみました。

7.1 レポート概要

- レポート名
 - Analysis of the Debian Developer Survey (2022) Learning more about possible uses of money in the Debian community.
 - レポートファイル <https://debian.pages.debian.net/dd-surveys/dd-survey-analysis-2022.pdf>
- レポートを周知するメール
 - <https://lists.debian.org/debian-devel-announce/2023/04/msg00001.html>
 - Date: Tue, 4 Apr 2023 19:43:56 -0400
 - Subject: Results of the Debian Developer's Survey about Usage of Money in Debian
- 回答者
 - Debian Developer の人
 - 266 人がアンケート回答の Web ページへアクセスし、実際の回答数は 224 人 (回答率 84 %)
 - * 2023 年度の DPL 選挙の投票資格をもつ人は 1023 人
 - * Debian Developer 全体で見ると回答率 22 %
 - レポートは 224 人の調査をまとめたもの
- 期間
 - 2022-04-07 回答呼びかけメール ^{*44}
 - 2023-04-03 レポート公開メール ^{*45}
- 結果概要
 - レポート PDF の冒頭、レポート公開メールに記述

7.2 レポートで気になった点

- P.4 DD の活動頻度、参加する作業

^{*44} <https://lists.debian.org/debian-devel-announce/2022/04/msg00002.html>

^{*45} <https://lists.debian.org/debian-devel-announce/2023/04/msg00001.html>

- P.10 どのメーリングリストを見ているか
- P.11 メール以外のコミュニケーション手段
- P.13 Debian の活動への参加は仕事と個人、また両方か
- P.16 今のやり方で Debian の活動を長期的に継続できそうか、また報酬があるとさらに継続できそうか
- P.20 Debian の活動にもっと時間をかけたいか
- P.24 Debian の活動が報酬につながっているか

- P.27 Grow Your Ideas プロジェクト^{*46}
 - 林健太郎さん @kenhys がアイデア募集の git をつくった^{*47}
 - P.29 実際に集まったアイデアと賛成、反対の意見
- P.34 Debian Project の寄付金を誰へ、何の作業へ抛出してよいかへの賛成、反対
 - P.43 回答者の所属チームによる傾向分析
- P.49 Debian の貢献者にお金を払うとした場合に課す条件

- P.51-85 具体的に何のアイデアの作業ならお金を払えるか (b1 から b13 のアイデア)
 - b3 infra improvements
 - b7 large changes
 - b4 Debian-specific software
 - b6 tech writers
 - b5 new service
 - b6 fund DPL
 - b1 pkg maint

- P.87-108 どの役割の人ならお金を払ってもいいか
 - a6 sec team
 - a7 LTS team
 - a3 RM-freeze (別の区分の RM-general は賛成が低い)
 - a1 DPL
 - a5 FTP master の中で NEW キューと RM リクエストを処理する人
- P.108 逆にどの役割の人にはお金を支払うべきではないか
 - Technical Committee (TC) のメンバー
 - Debian Account Manager のメンバー

- P.113 資金を提供するべきその他の役割 (自由記述)
 - DSA (Debian System Administrator)
 - 広報する人 (プレス、SNS、Web サイト、コミュニケーション)
 - 重要なパッケージやソフトウェアの管理者 (GNOME、dpkg など)
 - 法務や税務をサポートする人

- P.114-117 報酬を支払うことで Debian のボランティア文化に影響するか
- P.118-120 報酬を支払うことで起こる懸念 (自由記述)

^{*46} <https://salsa.debian.org/debian/grow-your-ideas>

^{*47} <https://lists.debian.org/debian-project/2021/04/msg00111.html>

8 2022 年を振り返って

2022 年 12 月の参加者



8.1 参加者

参加者は以下の 6 名でした。(敬称略)

- dictoss
- yy-y-ja-jp
- knok
- koedoyoshida
- ysaito
- jsynth21

8.2 参加者の意見

- Debian 勉強会の発表者が増えるとよい (少ない、固定化している)
- 2022 年は一度も Debian 勉強会をリアル開催できていない
 - リアル開催は年に数回でもいいかもしれない
 - 要望として、リアル開催する場合もオンライン参加できるようにしてほしいという話がある
 - リアル開催する会場でオンライン配信できるかの技術的な課題はある
 - * 回線については、モバイル回線で povo の 1 日使い放題や中容量契約など通信費が安価になってきたため、できそうな感じはある
 - オンラインでの勉強会は東京エリアと関西の合同開催という形にしているが、リアル開催の場合は開催日はそれぞれの個別の日程での開催になるのだろう (これまでは東京は毎月第 3 土曜日、関西は毎月第 4 日曜日の開催が多い)
 - * 関西でアテンドできる人がいるか?
- Debian 勉強会を開催していることを世間の人に気づけてもらえていないのではないか
 - メーリングリストで周知する時代でもない、と若手の参加者の方に言われたことがある
 - 若手の参加者は多くの場合は SNS で情報を得ている
 - * そのため、開催案内を SNS で流して多くの人の目に触れるようにする必要あり
 - 有料イベントの場合は、広報活動の一環として Twitter の有料広告を利用したことがある
 - * Twitter はフォローしている人だけに伝播するが、有料の広告だとフォロー外の人にも伝播される
 - * 自分が関わったイベントの経験の肌感覚では、SNS 経由の参加登録者は Facebook と Twitter では Twitter 経由で登録した方が Facebook の 2 倍いる

- IFTTT、Zatian などのツールは便利
- Instagram は未知数
- LINE はどうなんだろうか
- 2023/4/1 OSC 2023 Tokyo/Spring は物理開催する予定
 - 展示の案
 - * Debian のコミケ本
 - * Debian bookworm を入れたノート PC
 - * なにかインパクトある展示ができるか
 - ・ Steam Deck 展示する (手持ちの機械は OS が Arch Linux になってしまっているが)
 - ・ M1/M2 Mac を Debian arm64 で動かすデモをやれないか

8.2.1 各回の参加人数とセミナー

2022 年に実施した Debian 勉強会の参加人数^{*48}とセミナー内容は以下の通りです。

表 2 2022 年の勉強会の参加人数とセミナー

開催日	参加人数	セミナータイトル
2022-01-15	9 人	BoF「2021 年の振り返りと 2022 年の目標」
2022-02-19	20 人	「技術書典 12 で Debian の薄い本を頒布した話」 「Debian で始める Python Programming」
2022-03-19	8 人	「git-buildpackage を使ってみる」 「情報交換 & 交流タイム」
2022-04-16	8 人	「DDTP 及び DDTSS の紹介」 BoF「Debian のおすすめポイント」
2022-05-21	13 人	「Gmail とパスワード」 「アプリにおける多言語対応の仕組み」
2022-06-18	7 人	「相談・情報共有・もくもく会」
2022-07-18	9 人	「DebConf22 コソボ現地レポート」
2022-08-20	7 人	「DebConf 22 のイベント共有会」
2022-09-17	9 人	「ffmpeg + GIMP と Blender で動画編集 (CLI で呼吸している人向け)」
2022-10-15	9 人	「Debian で KVM を使う (CLI で呼吸している人向け)」 BoF「Debian 12 bookworm フリーズに向けた sid の日本語環境の動作確認状況共有」
2022-11-19	8 人	「Debian パッケージング repack 編 ~そのソフトウェア、そのまま配布して大丈夫だっけ~」
2022-12-17	6 人	BoF「2022 年を振り返って」 BoF「bookworm フリーズ直前課題共有及び相談」

^{*48} <https://salsa.debian.org/tokyodebian-team/tokyodebian-team.pages.debian.net/-/blob/master/event-history.txt>

本資料のライセンスについて

本資料はフリー・ソフトウェアです。あなたは、Free Software Foundation が公表した GNU GENERAL PUBLIC LICENSE の "バージョン 2" もしくはそれ以降が定める条項に従って本プログラムを再頒布または変更することができます。

本プログラムは有用とは思いますが、頒布にあたっては、市場性及び特定目的適合性についての暗黙の保証を含めて、いかなる保証も行ないません。詳細については GNU GENERAL PUBLIC LICENSE をお読みください。

ソースコードについて

本資料のソースコードは Git を使って <https://salsa.debian.org/tokyodebian-team/monthly-report.git> からダウンロードできます。以下に方法を示します。

```
$ sudo apt update
$ sudo apt install git
$ git clone https://salsa.debian.org/tokyodebian-team/monthly-report.git
```

GNU GENERAL PUBLIC LICENSE
Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.
51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

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 a brief idea of what it does.>
Copyright (C) <year> <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., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 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 Library General Public License instead of this License.

Debian オープンユースロゴ ライセンス

Copyright (c) 1999 Software in the Public Interest
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be

included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

『あんどきゅめんてっど でびあん』について

本書は、東京および関西周辺で毎月行なわれている『東京エリア Debian 勉強会』および『関西 Debian 勉強会』で使用された資料・小ネタ・必殺技などを一冊にまとめたものです。収録範囲は 2022/12 ~ 2023/05 まで。内容は無保証、つっこみなどがあれば勉強会にて。

是非、Debian 勉強会に下記の QRCode から参加してください!



あんどきゅめんてっど でびあん 2023 年冬号

2023 年 12 月 31 日 初版第 1 刷発行

東京エリア Debian 勉強会/関西 Debian 勉強会 (編集・印刷・発行)

Debian 勉強会 connpass: <https://debianjp.connpass.com>

<https://tokyodebian-team.pages.debian.net>

<https://wiki.debian.org/KansaiDebianMeeting>

