

debusine.debian.netを活用した パッケージの更新方法

しくみで"やらかし"アップロードを避ける

Kentaro Hayashi

ClearCode Inc.

2025年12月 東京エリア・関西合同Debian勉強会



- debusine.debian.netを活用したパッケージの更新方法
 - <https://slide.rabbit-shocker.org/authors/kenhys/tokyodebian-debusine-202512>



本日の内容

- ・Debusineとはなにか？
- ・Debusineで何ができるのか？
- ・Debusineに関する豆知識
- ・Debusineを使ってみよう



本日の内容

- ・Debusineとはなにか？
- ・Debusineで何ができるのか？
- ・Debusineに関する豆知識
- ・Debusineを使ってみよう

Debusineとはなにか？

Debusine

Log in

This Debusine instance is sponsored by [Freebox](#) for the benefit of the [Debian](#) project. It provides a range of packaging tools to Debian contributors and enables distribution-wide experiments. Check out [wiki.debian.org/DebusineDebianNet](#) to learn more about what's possible and how to use it.

If it's your first time exploring Debusine, you can start from [What is Debusine](#) for an introduction, or browse the index to the whole [Debusine documentation](#).

Any Debian Developer or Debian Maintainer can use this service by authenticating using their [salsa.debian.org](#) account.

Scopes and workspaces

Everything in Debusine happens inside [workspaces](#). Workspaces exist inside [scopes](#) which act as administrative dividers.

Scope	Workspace	Role	0	0	0
debian	base	Viewer	0	0	200
debian	developers	Viewer	394	327	1232
debian	develovers-creatives-test	Viewer	0	0	0
debian	develovers-test-crash	Viewer	0	0	0
debian	develovers-testy-django-test	Viewer	0	0	2
debian	develovers-testing-rcs-cmp	Viewer	0	0	0

© 2014-2017 Debusine Developers
Status: [Worker pool](#) • [Metrics](#) • [Task status](#) • [Theme: dark](#) • [Light](#) • [Info](#)
Debusine software project: [Documentation](#) • [Bugs](#) • [Code](#) • [Contributing](#) • 8.13.2.4ef 1019g0a0d764
[About debusine.debian.net](#) • [Documentation](#) • [Support](#) • [Sponsored by Freebox](#)

<https://debusine.debian.net>

debian



debusineの発音

DebConf25: Using Debusine to pre-test your unstable uploadsより



Debusineとはなにか？

Debusine's goal is to be an integrated solution to build, distribute and maintain a Debian-based distribution.



Debusineとはなにか？

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

- Freexianがスポンサーとなっているプロダクト
- パッケージング作業に関するインフラを提供する
- debusine.debian.netはDebusineを運用しているインスタンス
 - 外部の利用者からするとCIプラットフォーム



本日の内容

- ・Debusineとはなにか？
- ・**Debusine**で何ができるのか？
- ・Debusineに関する豆知識
- ・Debusineを使ってみよう



Debusineではなにができるの？

- ・DDやDMが利用できる (認証はsalsa.d.oと連携している)
 - **パッケージに関するCIをまるっとおまかせできる**
 - CI結果が問題なければ、そのままアップロードもできる
 - **tag2uploadとはまったく関係ない別のしくみ**
- tag2uploadについては2024年7月のDebian勉強会参照



それsalsa.d.oでよくない？

- ・現時点ではそれぞれメリット・デメリットがある
- ・debusineがsalsa.d.oのCIの完全な上位互換というわけでもない
 - あくまで2025年12月時点の話



salsa.d.oでのCI

- ・ビルド対象(amd64, i386)
- ・チェック対象
 - autopkgtest
 - blhc (hardeningのチェック)
 - debrebuild (**new!**)
 - lintian
 - piuparts



debusine.d.nでできるCI

- ・ビルド対象 (amd64, arm64, armhf, i386, ppc64el)
- ・チェック対象
 - autopkgtest
 - hardening (最近有効になったblhc)
 - lintian
 - piuparts



debusine.d.nとsalsa.d.oの比較

	debusine.d.n	salsa.d.o
ビルド対象arch	5	2 (amd64,i386)
チェック対象arch	5	amd64のみ
autopkgtest	✓	✓
blhc	✓	✓
debrebuild	✗	✓
lintian	✓	✓
piuparts	✓	✓



debusine.d.nとsalsa.d.o

・debusine.d.nがおすすめな人

- 複数アーキテクチャでチェックしたいDDもしくはDM
- CI完了までの速さが欲しい

・salsa.d.oがおすすめな人

- DDやDMではないコントリビューター
- salsa.d.oは利用者が多いので速度的に不利



本日の内容

- ・Debusineとはなにか？
- ・Debusineで何ができるのか？
- ・**Debusineに関する豆知識**
- ・Debusineを使ってみよう



ScopeとWorkspaces

- debusineではscopeとworkspacesが基本
- Scope:
 - 現在はdebianのみ (将来的にはdebian派生も考慮)
- Workspaces:
 - developersワークスペースをDD等が利用する



ScopeとWorkspaces

Scopes and workspaces

Everything in Debusine happens inside [workspaces](#). Workspaces exist inside [scopes](#) which act as administrative divisions.

Scope	Workspace	Role	R	S	C
debian	base	Viewer	0	0	202
debian	developers	Contributor	167	180	1321
debian	developers-cjwatson_test	Viewer	0	0	0
debian	developers-lamby_django_test	Viewer	0	0	2
debian	developers-roehling.robin_map	Viewer	0	0	0

© 2019-2025 [Debusine Developers](#)

参考: <https://debusine.debian.net>



Workflows

- ・どのチャンネルに対してパッケージング作業するかを示す
 - 通常はunstableにアップロードするのでupload-to-unstableを利用する
 - アップロードするソースパッケージで自動的に決まる



Workflowsの種類

Workflows

Most interaction with Debusine happens using [wf workflows](#).

The developers workspace currently has the following workflow templates available:

Name	Purpose
wf create-experiment	Create a workspace for an experiment
wf upload-to-experimental	Build, test, and upload to Debian experimental
wf upload-to-unstable	Build, test, and upload to Debian unstable
wf upload-to-trixie	Build, test, and upload to Debian trixie (via proposed-updates)
wf upload-to-trixie-security	Build, test, and upload to Debian trixie-security (only release in coordination with security team!)
wf upload-to-bookworm	Build, test, and upload to Debian bookworm (via bookworm-proposed-updates)
wf upload-to-bookworm-security	Build, test, and upload to Debian bookworm-security (only release in coordination with security team!)
wf upload-to-bullseye-security	Build, test, and upload to Debian bullseye-security (only release in coordination with LTS team!)

参考: <https://wiki.debian.org/DebusineDebianNet>





Workflowsの例

参考: <https://debusine.debian.net/debian/developers/workflow/>



本日の内容

- ・Debusineとはなにか？
- ・Debusineで何ができるのか？
- ・Debusineに関する豆知識
- ・**Debusineを使ってみよう**



Debusineを使えるようにする

- ・前提: debusineでCIを実行してunstableへアップロードする
 - アクセストークンの事前発行 (debusine.d.n)
 - debusineクライアントのインストール
 - セットアップを実行してトークンを設定



アクセストークンの事前発行(1)

Here you can generate and maintain the API tokens that are needed to [set up debusine-client](#) to access this instance of Debusine.

You can generate a token manually, or you can run `debusine-setup` to automatically register your client. You can remove tokens here to revoke access that you previously granted.

Here is an example `~/.config/debusine/client/config.ini` file you can use:

```
[General]
default-server = debusine.debian.net

[server:debusine.debian.net]
api-url = https://debusine.debian.net/api
scope = debian
token = generated-token-goes-here
```

Your existing tokens

You do not currently have any tokens: [create a new token](#) to be able to use debusine-client.

参考: [debusine.d.n](#)ユーザー設定のTokens画面



アクセストークンの事前発行(2)

Comment:
debusine

Enabled:

[Create](#)

参考: debusine.d.nでcreate a new tokenを実行した画面

アクセストークンの事前発行(3)

New token created.

Created at	2025-12-10 04:34:13
Enabled	✓
Comment	debusine
Token key	 

Save the token key: The server does not keep a copy. If you lose it you need to generate a new one.

[Token list](#)

参考: `create a new token`を実行した画面



debusineクライアントのインストール

- 必要なパッケージをインストール

```
$ sudo apt install dput-ng debusine-client
```



debusine setupを実行する

```
$ debusine setup
```



debusine setup (1)

```
> debusine setup
      Configured servers
      #  Name      Current  Description
      1  debian    debusine.debian.net
      2  freexian  Freexian (E)LTS
      3  localhost debusine-admin runserver for development

      1...3, new, quit:
```



debusine setup (2)

```
> 1
💡 Default server was not set, setting it to debian

Configuration for debian (default server)

[server:debian]
scope = debian          (scope)
api-url = https://debusine.debian.net/api/ (url)
token = acquired on save (token)
default-workspace = developers (workspace)
```

Commands: url, token, scope, (make) default, save, quit:



debusine setup (3)

```
> token
Enter (or clear) API token
> (あらかじめ生成しておいたトークンをここにコピペする)
```

```
Configuration for debian (default server)
```

```
[server:debian]
scope = debian          (scope)
api-url = https://debusine.debian.net/api/ (url)
token = present          (token)
default-workspace = developers (workspace)
```

```
Commands: url, token, scope, (make) default, save, quit:
```



debusine setup (4)

> save



Configuration saved



サンプル: Groongaの例

- ・前提: パッケージの修正はひととおり終わった状態

```
$ dput debusine.debian.net ./groonga_15.2.1+dfsg-1_source.changes
```



groonga: dput-ngでアップロードしたとき

```
Uploading groonga using debusine to debusine.debian.net (host: debusine.debian.net; directory: /)
running debusine-check-workflow: check debusine workflow for distribution
running checksum: verify checksums before uploading
running suite-mismatch: check the target distribution for common errors
running gpg: check GnuPG signatures before the upload
Not checking GPG signature due to allow_unsigned_uploads being set.
Uploading groonga_15.2.1+dfsg-1.dsc
Uploading groonga_15.2.1+dfsg.orig.tar.xz
Uploading groonga_15.2.1+dfsg-1.debian.tar.xz
Uploading groonga_15.2.1+dfsg-1_amd64.buildinfo
Uploading groonga_15.2.1+dfsg-1_source.changes
Uploading /home/kenhys/work/debian/groonga/build-area/groonga_15.2.1+dfsg-1_amd64.buildinfo to https://debusine.debian.net/api/1.0/artifact/2862652/files/groonga_15.2.1%2Bdfsg-1_amd64.buildinfo/
Uploading /home/kenhys/work/debian/groonga/build-area/groonga_15.2.1+dfsg-1.debian.tar.xz to https://debusine.debian.net/api/1.0/artifact/2862652/files/groonga_15.2.1%2Bdfsg-1.debian.tar.xz/
Uploading /home/kenhys/work/debian/groonga/build-area/groonga_15.2.1+dfsg-1.dsc to https://debusine.debian.net/api/1.0/artifact/2862652/files/groonga_15.2.1%2Bdfsg-1.dsc/
Uploading /home/kenhys/work/debian/groonga/build-area/groonga_15.2.1+dfsg-1_source.changes to https://debusine.debian.net/api/1.0/artifact/2862652/files/groonga_15.2.1%2Bdfsg-1_source.changes/
Uploading /home/kenhys/work/debian/groonga/build-area/groonga_15.2.1+dfsg.orig.tar.xz to https://debusine.debian.net/api/1.0/artifact/2862652/files/groonga_15.2.1%2Bdfsg.orig.tar.xz/
Created artifact: https://debusine.debian.net/debian/developers/artifact/2862652/
running debusine-create-workflow: create a debusine workflow
Created workflow: https://debusine.debian.net/debian/developers/work-request/259991/
```



groonga: ワークフロー開始



salsa.d.oのCIとdebusine.d.nの処理 の目安(1)

- ・10分でできること
 - salsa.d.o: sourceの展開くらい
 - debusine.d.n: arm や ppc64el を除いてビルド完走

パッケージや負荷状況で変わることに注意



salsa.d.oのCIとdebusine.d.nの処理の目安(2)

- ・20分でできること
 - salsa.d.o: まだビルド中
 - debusine.d.n: CI完走

パッケージや負荷状況で変わることに注意



salsa.d.oのCIとdebusine.d.nの処理 の目安(3)

- ・40分でできること
 - salsa.d.o: ようやくCIが完走する
- パッケージや負荷状況で変わることに注意



groonga: salsa.d.oでのCI

Debian | groonga | Kentaro Hagaishi created pipeline for commit 7e66c12 | 39 minutes ago, finished just now

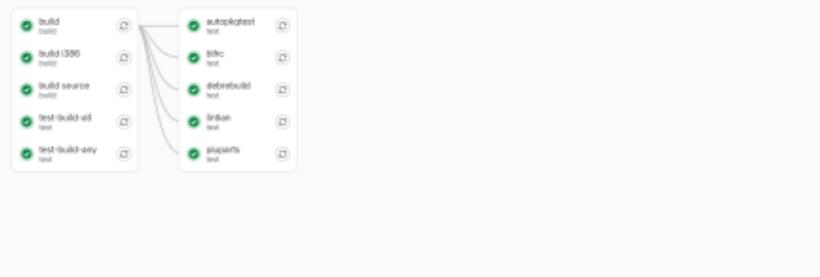
For [debainfinito](#)

[Start](#) [Branch](#) 60/10 jobs | 30 minutes 59 seconds, queued for 9 seconds

[Pipeline](#) [Jobs](#) [Tests](#)

Group jobs by [Stage](#) [Job dependencies](#) [Show dependencies](#)

Tip: Hover over a job to see the jobs it depends on to run.



build test
build (395) test
build source test
test-build-all test
test-build-any test
autopkgtest
lto test
databuild test
lindian test
pupans test

What's new

Help



groonga: ワークフロー完走したとき





groonga: パッケージに署名してアップロードする

The screenshot shows a workflow task titled "#260032: externaldebsign". The task is in a "Waiting for signature" state. It has two input artifacts: "debconf_appliance" and "package_externaldebsign". The "Output" tab is selected, showing the following output table:

Output	Value
externaldebsign	dh-25991
developers	kenhyo@gmail.com
kenhyo@gmail.com	26 minutes ago
26 minutes	3 minutes
26 minutes	2 months, 9 weeks

Below the table, there is a note: "Run 'debsign --server dejima_debsign_debian_provide --signature 260032' to sign this request. If you do not want to sign it, you can [abort](#) it instead."

© 2019-2023 Dejima Development
Status: Worker pools • Workers: 2048 cores • Thread limit: 8 • Light: auto
Dejima software project: Documentation • Bugs • Code • Contributing • 0.15.2-dev (1154 commits)
About dejima/dejima.net: Documentation • Support • Sponsorship/Community



groonga: 署名してアップロード実行

- 実行例: リクエストIDは都度変わる

```
debusine --server debusine.debian.net/debian provide-signature (リクエストID)
```



groonga: 署名してアップロード実行

```
debusine --server debusine.debian.net/debian provide-signature 260032
Artifact file downloaded: /tmp/debusine-debsign-_zgpdf/groonga_15.2.1+dfsg-1_amd64.buildinfo
Artifact file downloaded: /tmp/debusine-debsign-_zgpdf/groonga_15.2.1+dfsg-1.dsc
Artifact file downloaded: /tmp/debusine-debsign-_zgpdf/groonga_15.2.1+dfsg-1_source.changes
signfile dsc groonga_15.2.1+dfsg-1.dsc 719EB2D93DBE9C4D21FBA064F7FB75C566ED20E3

fixup_buildinfo groonga_15.2.1+dfsg-1.dsc groonga_15.2.1+dfsg-1_amd64.buildinfo
signfile buildinfo groonga_15.2.1+dfsg-1_amd64.buildinfo 719EB2D93DBE9C4D21FBA064F7FB75C566ED20E3

fixup_changes dsc groonga_15.2.1+dfsg-1.dsc groonga_15.2.1+dfsg-1_source.changes
fixup_changes buildinfo groonga_15.2.1+dfsg-1_amd64.buildinfo groonga_15.2.1+dfsg-1_source.changes
signfile changes groonga_15.2.1+dfsg-1_source.changes 719EB2D93DBE9C4D21FBA064F7FB75C566ED20E3

Successfully signed dsc, buildinfo, changes files
Uploading /tmp/debusine-debsign-_zgpdf/groonga_15.2.1+dfsg-1_amd64.buildinfo to https://debusine.debian.net/api/1.0/artifact/2862808/files/groonga_15.2.1%2Bdfsg-1_amd64.buildinfo/
Uploading /tmp/debusine-debsign-_zgpdf/groonga_15.2.1+dfsg-1.dsc to https://debusine.debian.net/api/1.0/artifact/2862808/files/groonga_15.2.1%2Bdfsg-1.dsc/
Uploading /tmp/debusine-debsign-_zgpdf/groonga_15.2.1+dfsg-1_source.changes to https://debusine.debian.net/api/1.0/artifact/2862808/files/groonga_15.2.1%2Bdfsg-1_source.changes/
```



さいごに

- ・ DDやDMであれば、debusine.d.nを併用するのがおすすめ
- ・ 事前にdebusine.d.nでCIを実行することで特定の環境の問題に気づける
- ・DDやDMになろう！