

# deb パッケージの操作と構築①

2014.5.26

R&Dセンター

OSS戦略企画室 OSS技術第二課

※本文中の会社名、商品名は、各社の商標及び登録商標です。

# パッケージ管理の種類

パッケージは対象とするソフトの必要なライブラリ、設定ファイルをまとめて配布する形式。

Linux ではディストリビューションによって各々独自のパッケージ管理を行っている。現状では主に下記の種類がある。

- Slackware: tar ball 形式

slackware, SUSE, Plamo , ...

- Debian系: deb 形式

Debian, Ubuntu, Knopix, ...

- RedHat系: rpm形式

RedHat, Vine, TurboLinux, SUSE, ...

以降では Ubuntu 13.10 環境でのパッケージ管理について説明する。

## Ubuntu のバージョン

- 年4月,10月の半年毎のリリース。
- バージョン番号は  
Ubuntu {release 年}.{release 月}
- コードネームはアルファベット順
- lsb-release コマンドでバージョンを確認できる  
(lsb は Linux Standard Base の意味)

下記パッケージが必要。

Ubuntu: lsb-release

RedHat: redhat-lsb

## ➤ Ubuntu でのバージョン確認

```
$ lsb_release -a
No LSB modules are available.
Distributor ID: Ubuntu
Description:   Ubuntu 13.10
Release:      13.10
Codename:     saucy
$
```

### 直接ファイルを見る場合

```
$ cat /etc/lsb-release
DISTRIB_ID=Ubuntu
DISTRIB_RELEASE=13.10
DISTRIB_CODENAME=saucy
DISTRIB_DESCRIPTION="Ubuntu 13.10"
$

$ cat /etc/os-release
NAME="Ubuntu"
VERSION="13.10, Saucy Salamander"
ID=ubuntu
ID_LIKE=debian
PRETTY_NAME="Ubuntu 13.10"
VERSION_ID="13.10"
HOME_URL="http://www.ubuntu.com/"
SUPPORT_URL="http://help.ubuntu.com/"
BUG_REPORT_URL="http://bugs.launchpad.net/ubuntu/"
$
```

## ➤ RedHatでのバージョン確認

```
$ lsb_release -a
LSB Version: :core-4.0-amd64:core-4.0-noarch:graphics-4.0-amd64:graphics-4.0-noarch:printing-4.0-
amd64:printing-4.0-noarch
Distributor ID: Scientific
Description: Scientific Linux release 6.1 (Carbon)
Release: 6.1
Codename: Carbon
$
```

## 直接ファイルを見る場合

```
$ cat /etc/redhat-release
Scientific Linux release 6.1 (Carbon)
$
```

# パッケージの種別

## パッケージには2種類ある

- バイナリパッケージ

実行ファイル、ライブラリを格納するインストール用パッケージ。  
アーキテクチャ毎に存在する。

```
package-name-{software-version}-{ubuntu-version}_{arch}.{deb | udeb}
```

- ソースパッケージ

パッケージビルドの為に使用する upstream のソースファイル、  
dsc ファイル、差分ファイル。

```
package-name-{software-version}-{ubuntu-version}.dsc
```

```
package-name-{software-version}-{ubuntu-version}.debian.tar.gz
```

```
[package-name-{software-version}.orig.tar.gz]
```

➤ deb ソースパッケージ例

```
$ apt-get source hello
```

```
...
```

```
$ ls -ACF1
```

```
hello-2.7/
```

```
hello_2.7-2.debian.tar.gz
```

```
hello_2.7-2.dsc
```

```
hello_2.7.orig.tar.gz
```

```
$
```

# パッケージ操作

- Debian系のディストリビューションで使用するパッケージ操作コマンド

dpkg、apt-get、apt-cache

(dselect, aptitude 等は別途機会があれば紹介)

- dpkg

単純なパッケージ操作コマンド

- パッケージは自分で用意する必要がある。
- dpkg のパッケージインストールではパッケージファイルを直接指定する。
- 依存関係は自分で解決する必要がある。



- apt-get

- 依存関係を解決してくれる。
- ネットワークリポジトリ上のパッケージを参照できる。

- apt-cache

- レポジトリ上のパッケージを検索できる。
- パッケージの情報を参照できる
  - パッケージの概要
  - パッケージの依存関係

# dpkgコマンドでの操作

## インストール関連の操作

機能	コマンド
新規インストール。 既インストール済みのパッケージのアップグレード 同バージョンのパッケージの再インストール	<code>dpkg -i package-file-name</code>
パッケージのダウングレード	<code>dpkg --force-downgrade -i package-file-name</code>
パッケージのアンインストール(設定ファイルは残る)	<code>dpkg -r --remove package-name</code>
パッケージのアンインストール(全て削除)	<code>dpkg -P --purge package-name</code>
インストール失敗時の修復	<code>dpkg --configure package-name</code> <code>dpkg --configure -a</code>

## インストール操作時の注意点

- インストールする際には対象とするパッケージの依存関係を満たす必要がある。必要となるパッケージは予めインストールしておくか対象パッケージと一緒にインストールする必要がある。
- インストール途中で `postscript` 等が失敗した場合 `-configure` オプションで修復できる場合がある。

## ➤ インストール、再インストールと削除操作の例

```
$ dpkg -l | grep hello
$ sudo dpkg -i hello_2.4-3_amd64.deb
Selecting previously unselected package hello.
(Reading database ... 95894 files and directories currently installed.)
Unpacking hello (from hello_2.4-3_amd64.deb) ...
Setting up hello (2.4-3) ...
Processing triggers for install-info ...
Processing triggers for man-db ...
$ dpkg -l | grep hello
ii hello                2.4-3                amd64    The classic greeting, and a good
example
$ sudo dpkg -i hello_2.4-3_amd64.deb
(Reading database ... 95902 files and directories currently installed.)
Preparing to replace hello 2.4-3 (using hello_2.4-3_amd64.deb) ...
Unpacking replacement hello ...
Setting up hello (2.4-3) ...
Processing triggers for install-info ...
Processing triggers for man-db ...
$ sudo dpkg -r hello
(Reading database ... 95902 files and directories currently installed.)
Removing hello ...
Processing triggers for man-db ...
Processing triggers for install-info ...
$ dpkg -l | grep hello
$
```

## ➤ アップデートとダウングレード操作の例

```
$ dpkg -l | grep hello
ii hello                2.4-3                amd64    The classic greeting, and a good
example
$ sudo dpkg -i hello_2.7-2_amd64.deb
(Reading database ... 95902 files and directories currently installed.)
Preparing to replace hello 2.4-3 (using hello_2.7-2_amd64.deb) ...
Unpacking replacement hello ...
Setting up hello (2.7-2) ...
Processing triggers for install-info ...
Processing triggers for man-db ...
$ dpkg -l | grep hello
ii hello                2.7-2                amd64    The classic greeting, and a good
example
$ sudo dpkg --force-downgrade -i hello_2.4-3_amd64.deb
dpkg: warning: downgrading hello from 2.7-2 to 2.4-3
(Reading database ... 95901 files and directories currently installed.)
Preparing to replace hello 2.7-2 (using hello_2.4-3_amd64.deb) ...
Unpacking replacement hello ...
Setting up hello (2.4-3) ...
Processing triggers for install-info ...
Processing triggers for man-db ...
$ dpkg -l | grep hello
ii hello                2.4-3                amd64    The classic greeting, and a good
example
$
```

## インストールコマンドの併用オプション

機能	コマンド
既インストール済みのパッケージをインストールしない	-E --skip-same-version
指定ディレクトリ下の全パッケージをインストール	-R --recursive
パッケージの衝突を無視する	--force-conflicts
パッケージの依存関係を無視する	--force-depends
インストールパッケージより新しいパッケージがインストール済みでも実行	--force-downgrade
指定パッケージの依存関係を無視する	--ignore-depends=package-name
インストールディレクトリを指定ディレクトリに変更	--instdir=dir
/var/lib/dpkg の代わりに指定ディレクトリを使用	--admindir=dir
--instdir=dir, --admindir=dir/var/lib/dpkg と同じ	--root=dir

# パッケージ情報の参照

機能	コマンド
パッケージファイルの詳細情報の表示	<code>dpkg -l --info package-file-name</code>
パッケージファイルの格納ファイルの表示。	<code>dpkg -c --contents package-file-name</code>
インストールパッケージの詳細情報の表示	<code>dpkg -s --status package-name</code>
インストールパッケージの所属ファイルの表示。	<code>dpkg -L --listfiles package-name</code>
パッケージの一覧	<code>dpkg -l --list</code>
所属パッケージの表示	<code>dpkg -S --search file-path</code>

## ➤ パッケージファイルの詳細情報表示の例

```
$ dpkg -I hello_2.4-3_amd64.deb
new debian package, version 2.0.
size 34266 bytes: control archive=585 bytes.
    713 bytes, 17 lines   control
Package: hello
Version: 2.4-3
Architecture: amd64
Maintainer: Ubuntu Developers <ubuntu-devel-discuss@lists.ubuntu.com>
Original-Maintainer: Santiago Vila <sanvila@debian.org>
Installed-Size: 648
Depends: libc6 (>= 2.3.4), dpkg (>= 1.15.4) | install-info
Section: devel
Priority: optional
Description: The classic greeting, and a good example
 The GNU hello program produces a familiar, friendly greeting. It
 allows non-programmers to use a classic computer science tool which
 would otherwise be unavailable to them.
.
 Seriously, though: this is an example of how to do a Debian package.
 It is the Debian version of the GNU Project's `hello world' program
 (which is itself an example for the GNU Project).
$
```



## ➤ パッケージファイルの詳細情報表示の例(続き)

```
$ dpkg -s hello
Package: hello
Status: install ok installed
Priority: optional
Section: devel
Installed-Size: 648
Maintainer: Ubuntu Developers <ubuntu-devel-discuss@lists.ubuntu.com>
Architecture: amd64
Version: 2.4-3
Depends: libc6 (>= 2.3.4), dpkg (>= 1.15.4) | install-info
Description: The classic greeting, and a good example
The GNU hello program produces a familiar, friendly greeting. It
allows non-programmers to use a classic computer science tool which
would otherwise be unavailable to them.
.
Seriously, though: this is an example of how to do a Debian package.
It is the Debian version of the GNU Project's `hello world' program
(which is itself an example for the GNU Project).
Original-Maintainer: Santiago Vila sanvila@debian.org
$
```

## ➤ パッケージファイルに所属のファイルの表示の例

```
$ dpkg -c hello_2.4-3_amd64.deb
drwxr-xr-x root/root      0 2009-11-05 17:17 ./
drwxr-xr-x root/root      0 2009-11-05 17:17 ./usr/
drwxr-xr-x root/root      0 2009-11-05 17:17 ./usr/share/
drwxr-xr-x root/root      0 2009-11-05 17:17 ./usr/share/doc/
drwxr-xr-x root/root      0 2009-11-05 17:17 ./usr/share/doc/hello/
-rw-r--r-- root/root    2645 2008-12-10 03:48 ./usr/share/doc/hello/NEWS
-rw-r--r-- root/root    2246 2009-11-05 17:16 ./usr/share/doc/hello/copyright
-rw-r--r-- root/root    6420 2008-12-10 03:49 ./usr/share/doc/hello/changelog.gz
-rw-r--r-- root/root    3472 2009-11-05 17:16 ./usr/share/doc/hello/changelog.Debian.gz
drwxr-xr-x root/root      0 2009-11-05 17:17 ./usr/share/info/
-rw-r--r-- root/root   11421 2009-11-05 17:17 ./usr/share/info/hello.info.gz
drwxr-xr-x root/root      0 2009-11-05 17:17 ./usr/share/man/
drwxr-xr-x root/root      0 2009-11-05 17:17 ./usr/share/man/man1/
-rw-r--r-- root/root     687 2009-11-05 17:17 ./usr/share/man/man1/hello.1.gz
drwxr-xr-x root/root      0 2009-11-05 17:17 ./usr/bin/
-rwxr-xr-x root/root   18728 2009-11-05 17:17 ./usr/bin/hello
$
```

- インストールパッケージの所属ファイルの表示例  
所属パッケージ検索の際のファイルは絶対パスで指定する

```
$ dpkg -L hello
/.
/usr
/usr/share
/usr/share/doc
/usr/share/doc/hello
/usr/share/doc/hello/NEWS
/usr/share/doc/hello/copyright
/usr/share/doc/hello/changelog.gz
/usr/share/doc/hello/changelog.Debian.gz
/usr/share/info
/usr/share/info/hello.info.gz
/usr/share/man
/usr/share/man/man1
/usr/share/man/man1/hello.1.gz
/usr/bin
/usr/bin/hello
$ dpkg -S /usr/bin/hello
hello: /usr/bin/hello
$
```

# バイナリパッケージの展開

deb パッケージは ar 形式で一つにまとめている。  
その為中身を確認したい場合は ar コマンドにより展開  
できる。

## ➤ パッケージの展開

```
$ ls
hello_2.4-3_amd64.deb
$ ar -x hello_2.4-3_amd64.deb
$ ls
control.tar.gz data.tar.gz debian-binary hello_2.4-3_amd64.deb
$ tar xzf control.tar.gz
$ tarxzf data.tar.gz
tarxzf: command not found
$ tar xzf data.tar.gz
$ ls
control control.tar.gz data.tar.gz debian-binary hello_2.4-3_amd64.deb usr
$
```

## ➤ パッケージの展開(続き)

```
$ cat debian-binary
2.0
$ cat control
Package: hello
Version: 2.4-3
Architecture: amd64
Maintainer: Ubuntu Developers <ubuntu-devel-discuss@lists.ubuntu.com>
Original-Maintainer: Santiago Vila <sanvila@debian.org>
Installed-Size: 648
Depends: libc6 (>= 2.3.4), dpkg (>= 1.15.4) | install-info
Section: devel
Priority: optional
Description: The classic greeting, and a good example
The GNU hello program produces a familiar, friendly greeting. It
allows non-programmers to use a classic computer science tool which
would otherwise be unavailable to them.
.
Seriously, though: this is an example of how to do a Debian package.
It is the Debian version of the GNU Project's `hello world' program
(which is itself an example for the GNU Project).
$
```

## ➤ パッケージの展開(続き)

```
$ ls -R usr
usr:
bin share

usr/bin:
hello

usr/share:
doc info man

usr/share/doc:
hello

usr/share/doc/hello:
NEWS changelog.Debian.gz changelog.gz copyright

usr/share/info:
hello.info.gz

usr/share/man:
man1

usr/share/man/man1:
hello.1.gz
$
```

# apt コマンドでの操作

## インストール関連の操作

機能	コマンド
新規インストール	<code>apt-get install package-name</code>
パッケージの再インストール	<code>apt-get --reinstall install package-name</code>
指定パッケージのインストール	<code>apt-get install package-name=version</code>
インストール動作の確認	<code>apt-get -s   --simulate install package-name</code>
パッケージのアンインストール(設定ファイルは残る)	<code>apt-get remove package-name</code>
パッケージのアンインストール(全ファイルの削除)	<code>apt-get purge package-name</code>

## ➤ インストール関連操作の例

```
$ sudo dpkg -i hello_2.9-1_amd64.deb
Selecting previously unselected package hello.
(Reading database ... 95934 files and directories currently installed.)
Unpacking hello (from hello_2.9-1_amd64.deb) ...
Setting up hello (2.9-1) ...
Processing triggers for install-info ...
Processing triggers for man-db ...
$ dpkg -l | grep hello
ii hello                2.9-1                amd64        example package based on GNU hello
$ sudo apt-get install hello
Reading package lists... Done
Building dependency tree
Reading state information... Done
hello is already the newest version.
0 upgraded, 0 newly installed, 0 to remove and 96 not upgraded.
$ sudo apt-get install hello=2.8-4
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages will be DOWNGRADED:
 hello
0 upgraded, 0 newly installed, 1 downgraded, 0 to remove and 96 not upgraded.
Need to get 0 B/28.1 kB of archives.
After this operation, 4096 B of additional disk space will be used.
Do you want to continue [Y/n]? y
```



## ➤ インストール関連操作の例(続き)

```
dpkg: warning: downgrading hello from 2.9-1 to 2.8-4
(Reading database ... 95941 files and directories currently installed.)
Preparing to replace hello 2.9-1 (using ../archives/hello_2.8-4_amd64.deb) ...
Unpacking replacement hello ...
Processing triggers for install-info ...
Processing triggers for man-db ...
Setting up hello (2.8-4) ...
$ dpkg -l | grep hello
ii hello                2.8-4                amd64        The classic greeting, and a good example
$ sudo apt-get purge hello
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages will be REMOVED:
 hello*
0 upgraded, 0 newly installed, 1 to remove and 96 not upgraded.
After this operation, 111 kB disk space will be freed.
Do you want to continue [Y/n]? y
(Reading database ... 95941 files and directories currently installed.)
Removing hello ...
Processing triggers for man-db ...
Processing triggers for install-info ...
$ dpkg -l | grep hello
$
```

## インストール関連の操作(続き)

機能	コマンド
全パッケージのアップデート(現状維持)	apt-get upgrade
全パッケージのアップデート(パッケージの調整を行う)	apt-get dist-upgrade
レポジトリ情報の更新	apt-get update

- upgrade

現状のインストールパッケージのバージョンアップのみ行う。  
依存関係により現状の変更が必要となる場合は、そのパッケージのバージョンアップはスキップする。

- dist-upgrade

依存関係に応じ必要なパッケージの新規インストール、削除を行う。

## インストール関連の操作(続き)

- 特定のパッケージを upgrade から筈する場合  
dpkg コマンドによりパッケージを hold できる。  
    echo package-name hold | dpkg --set-selections  
状態は下記のコマンドにより確認できる。  
    dpkg --get-selections  
hold を解除する場合  
    echo package-name install | dpkg --set-selections

# パッケージ情報の参照

機能	コマンド
パッケージの検索	<code>apt-cache [--names-only] search key-word</code>
パッケージの詳細情報の表示	<code>apt-cache show package-name</code>
パッケージの依存情報の表示	<code>apt-cache showpkg package-name</code>
パッケージの一覧	<code>apt-cache pkgnames [key-word]</code>

- `--names-only`  
パッケージ名に `key-word` が含まれるもののみリストする
- `pkgnames key-word`  
`key-word` が指定された場合、先頭が `key-word` に一致するパッケージのみ表示する。

## ➤ パッケージ情報表示の例

```
$ apt-cache show hello
Package: hello
Priority: optional
Section: devel
Installed-Size: 108
Maintainer: Ubuntu Developers <ubuntu-devel-discuss@lists.ubuntu.com>
Original-Maintainer: Santiago Vila <sanvila@debian.org>
Architecture: amd64
Version: 2.8-4
Depends: libc6 (>= 2.14), dpkg (>= 1.15.4) | install-info
Filename: pool/main/h/hello/hello_2.8-4_amd64.deb
Size: 28148
MD5sum: af651f8b526976269affa5b888e63db7
SHA1: c353f6e99dad0dc628b5604741bfbf97eaf74bcf
SHA256: 0cd79de61001c8f6ab7611ba724f0f7da7fd0998bf0fd98710fe26d81f0a6b18
Description-en: The classic greeting, and a good example
The GNU hello program produces a familiar, friendly greeting. It
allows non-programmers to use a classic computer science tool which
would otherwise be unavailable to them.
.
Seriously, though: this is an example of how to do a Debian package.
It is the Debian version of the GNU Project's `hello world' program
(which is itself an example for the GNU Project).
Description-md5: b7df6fe7ffb325083a3a60819a7df548
Homepage: http://www.gnu.org/software/hello/
Description-md5: b7df6fe7ffb325083a3a60819a7df548
Bugs: https://bugs.launchpad.net/ubuntu/+filebug
Origin: Ubuntu
Supported: 9m
```

```
$
```

## ➤ パッケージの依存情報表示の例

```
$ apt-cache showpkg hello
Package: hello
Versions:
2.8-4 (/var/lib/apt/lists/jp.archive.ubuntu.com_ubuntu_dists_saucy_main_binary-amd64_Packages)
(/var/lib/dpkg/status)
Description Language:
    File: /var/lib/apt/lists/jp.archive.ubuntu.com_ubuntu_dists_saucy_main_binary-amd64_Packages
    MD5: b7df6fe7ffb325083a3a60819a7df548
Description Language: en
    File: /var/lib/apt/lists/jp.archive.ubuntu.com_ubuntu_dists_saucy_main_i18n_Translation-en
    MD5: b7df6fe7ffb325083a3a60819a7df548

Reverse Depends:
hello-debhelper:i386,hello
hello-debhelper:i386,hello
hello:i386,hello
junior-system,hello
hello-debhelper,hello
hello-debhelper,hello
Dependencies:
2.8-4 - libc6 (2 2.14) dpkg (18 1.15.4) install-info (0 (null)) hello:i386 (0 (null))
Provides:
2.8-4 -
Reverse Provides:
hello-debhelper 2.8-1
$
```

## ➤ パッケージ一覧の例

```
$ apt-cache pkgnames  
account-plugin-yahoojp  
ceph-fuse  
dvd+rw-tools  
e3  
gnome-commander-data  
grub-gfxpayload-lists  
gweled  
libannotation-indexer-java-doc  
libboost-timer-dev  
libdune-istl-dev  
libfile-keepass-perl  
libghc-configfile-dev  
libgnome-desktop-3-dev  
libphat-dev  
libsimple-http-java-doc  
libsslcommon2-dev  
libtag1-dev  
moby  
pxlib1  
python3-doc  
python3-tz  
...  
$ apt-cache pkgnames iscsi  
iscsitarget  
iscsitarget-dkms  
$
```

## その他の操作

機能	コマンド
バイナリパッケージのダウンロード	apt-get download package-name apt-get -d --download-only install package-name
ソースパッケージのダウンロード	apt-get source package-name
パッケージファイルのクリア	apt-get clean
不要パッケージファイルのクリア	apt-get autoclean

- clean  
apt-get によりインストール、ダウンロードしたバイナリパッケージは /var/cache/apt/archives/ に格納される。必要無い場合は、このオプションにより削除できる。  
尚ソースパッケージはカレントディレクトリに格納される。
- autoclean  
古いパッケージファイルやシステムに不要なファイルを削除する



# パッケージのレポジトリ

## apt コマンドの設定

- レポジトリ情報

`/etc/apt/sources.list`

apt コマンドが参照するレポジトリを指定。

ファイルの各行は apt-line と呼ばれる。

- 環境設定

`/etc/apt/apt.conf.d/*`

apt の動作環境を設定。apt コマンドのオプションの default 値の設定等が行える。今回は説明しない。

詳しくは `man apt.conf` 参照

## ➤ apt-line の例

```
$ cat /etc/apt/sources.list
#

# deb cdrom:[Ubuntu-Server 13.10 _Saucy Salamander_ - Release amd64 (20131016)]/ saucy main
restricted

#deb cdrom:[Ubuntu-Server 13.10 _Saucy Salamander_ - Release amd64 (20131016)]/ saucy main
restricted

# See http://help.ubuntu.com/community/UpgradeNotes for how to upgrade to
# newer versions of the distribution.
deb http://jp.archive.ubuntu.com/ubuntu/ saucy main restricted
deb-src http://jp.archive.ubuntu.com/ubuntu/ saucy main restricted

## Major bug fix updates produced after the final release of the
## distribution.
deb http://jp.archive.ubuntu.com/ubuntu/ saucy-updates main restricted
deb-src http://jp.archive.ubuntu.com/ubuntu/ saucy-updates main restricted

## N.B. software from this repository is ENTIRELY UNSUPPORTED by the Ubuntu
## team. Also, please note that software in universe WILL NOT receive any
## review or updates from the Ubuntu security team.
deb http://jp.archive.ubuntu.com/ubuntu/ saucy universe
deb-src http://jp.archive.ubuntu.com/ubuntu/ saucy universe
deb http://jp.archive.ubuntu.com/ubuntu/ saucy-updates universe
deb-src http://jp.archive.ubuntu.com/ubuntu/ saucy-updates universe
...
```

## ● apt-line

下記の形式で指定する。

```
deb http://url-path/dir-name dist-version component
```

```
deb-src http://url-path/dir-name dist-version component
```

```
deb file:directory-path
```

```
deb-src file:directory-path
```

http, file 以外にも ftp, cdrom 等も指定できる

deb: バイナリパッケージの取得先

deb-src: ソースパッケージの取得先

dist-version: {ubuntu のコードネーム}-{種別}

例えば saucy, saucy-updates, saucy-security

component: 格納パッケージの区分を示す。  
下記の区分がある。

	フリー	ノンフリー
サポート有り	main	restricted
サポート無し	universe	multiverse

restricted は主にデバイスドライバー関連のパッケージを含む。

## ● リポジトリのツリー構成

```
dir-base +- dists +- saucy +- main +- binary-amd64 +- Packages.gz
|
|
|
| +- binary-i386 - ...
|
| +- source +- Sources.gz
|
| +- restricted +- binary-amd64
|
| ...
| +- Release, Release.gpg
| +- saucy-updates +- main
|
| ...
+- pool +- main
|
| +- multiverse
|
| +- restricted
|
| +- universe
```

- Packages

- バイナリパッケージの情報と MD5サムを格納

- Sources

- ソースパッケージの情報と MD5サムを格納

- Release

- Packages, Sources の MD5サムを格納

## ●リポジトリの作成

ローカルリポジトリ作成の簡単な例を示す。  
下記のディレクトリ構成を考える。

```
/data1/ubuntu/tests/repo
|
+- dists +- binary-amd64 +- Packages.gz
|         |                 +- Release
|         |                 +- Release.gpg
|         +- source +- Sources.gz
|                   +- Release
|                   +- Release.gpg
|
+- packages +- hello_2.4-3 – source & binary package
|           |
|           +- hello_2.7-2 – source & binary package
|           |
|           +- hello_2.8-2 – source & binary package
```

- 認証キーの作成

```
$ gpg -gen-key
途中省略
$ ls ~/.gnupg
gpg.conf private-keys-v1.d pubring.gpg pubring.gpg~ random_seed secring.gpg
trustdb.gpg
$ gpg --list-keys
/home/kakuma/.gnupg/pubring.gpg
-----
pub 2048R/07A608A4 2014-05-11
uid      fumihiko kakuma <kakuma@valinux.co.jp>
sub 2048R/C6EBD3BA 2014-05-11

$
```



## ● 認証キーの登録

```
$ gpg --export 07A608A4 | sudo apt-key add -
OK
$ apt-key list
/etc/apt/trusted.gpg
-----
pub 1024D/437D05B5 2004-09-12
uid          Ubuntu Archive Automatic Signing Key <ftpmaster@ubuntu.com>
sub 2048g/79164387 2004-09-12

pub 1024D/FBB75451 2004-12-30
uid          Ubuntu CD Image Automatic Signing Key <cdimage@ubuntu.com>

pub 4096R/C0B21F32 2012-05-11
uid          Ubuntu Archive Automatic Signing Key (2012) <ftpmaster@ubuntu.com>

pub 4096R/EFE21092 2012-05-11
uid          Ubuntu CD Image Automatic Signing Key (2012) <cdimage@ubuntu.com>

pub 2048R/07A608A4 2014-05-11
uid          fumihiko kakuma <kakuma@valinux.co.jp>
sub 2048R/C6EBD3BA 2014-05-11

$
```

- ソースパッケージ情報の作成

```
$ cd /data1/ubuntu/tests/repo/dists/source
/data1/ubuntu/tests/repo/dists/source$ apt-ftparchive sources ../../packages >
Sources
hello has no source override entry
hello has no binary override entry either
hello has no source override entry
hello has no binary override entry either
hello has no source override entry
hello has no binary override entry either
/data1/ubuntu/tests/repo/dists/source$ gzip -c9 Sources > Sources.gz
/data1/ubuntu/tests/repo/dists/source$ apt-ftparchive release . > Release
/data1/ubuntu/tests/repo/dists/source$ gpg -b -o Release.gpg Release
```

```
You need a passphrase to unlock the secret key for
user: "fumihiko kakuma <kakuma@valinux.co.jp>"
2048-bit RSA key, ID 07A608A4, created 2014-05-11
```

```
gpg: gpg-agent is not available in this session
/data1/ubuntu/tests/repo/dists/source $
```

## ● バイナリパッケージ情報の作成

```
/data1/ubuntu/tests/repo/dists/source$ cd ../binary-amd64/  
/data1/ubuntu/tests/repo/dists/binary-amd64$ apt-ftparchive packages ../../packages  
> Packages  
/data1/ubuntu/tests/repo/dists/binary-amd64$ gzip -c9 Packages > Packages.gz  
/data1/ubuntu/tests/repo/dists/binary-amd64$ gpg -b -o Release.gpg Release
```

You need a passphrase to unlock the secret key for  
user: "fumihiko kakuma <kakuma@valinux.co.jp>"  
2048-bit RSA key, ID 07A608A4, created 2014-05-11

```
gpg: gpg-agent is not available in this session  
/data1/ubuntu/tests/repo/dists/binary-amd64$ cd ..  
/data1/ubuntu/tests/repo$ ls dists/binary-amd64/  
Packages Packages.gz Release Release.gpg  
/data1/ubuntu/tests/repo$ ls dists/source/  
Release Release.gpg Sources Sources.gz  
/data1/ubuntu/tests/repo$
```

## ● パッケージ情報ファイルの内容

```
/data1/ubuntu/tests/repo$ cat dists/source/Sources
Package: hello
Binary: hello
Version: 2.4-3
Maintainer: Santiago Vila <sanvila@debian.org>
Architecture: any
Standards-Version: 3.8.3
Format: 1.0
Directory: ../../packages/hello_2.4-3
Files:
801470294f5fe9e38de9ab3a38761a3d 1174 hello_2.4-3.dsc
1691faa758ca41c70b6da5501bdf230a 499638 hello_2.4.orig.tar.gz
b8e1fdccc39cf9f15dd3c2d69c817b7b 5684 hello_2.4-3.diff.gz
Checksums-Sha1:
b571a35fb081253b6654cd8bf4f1ba20e0d6ec68 1174 hello_2.4-3.dsc
fcbf0264928900adf03a7797474375e1a6fa3836 499638 hello_2.4.orig.tar.gz
81f0c1762b30ae141f04145d1195a6f31b04a38b 5684 hello_2.4-3.diff.gz
Checksums-Sha256:
9f3fc36491ea6450c06ac7a43a119a64b05aa432e8c46497e5d4654334349b38 1174
hello_2.4-3.dsc
```

## ● パッケージ情報ファイルの内容(続き)

534745c4b7e063f5eb5f984609caf0f7c06d46df03e4d404f20996d28b6df1f7 499638

hello\_2.4.orig.tar.gz

bb6078fbced7ba3b6d7a2a890702181242005e9859d06a5517480c55f16091aa 5684

hello\_2.4-3.diff.gz

Checksums-Sha512:

b5b5e68f138e17a3c0dfb6938319a0772e3d67b6eb49741b61d2af3e45fd46762b11828

2b220035389a12907e2ad6ae9fac67ddeb58829060e418943a51d27d5 1174 hello\_2.4-3.dsc

bda3bd88bc02dee026a200964b110250d15e84b5ef107c0ce02d347c152596740a40fdfd8f3fd6d72c75ddcc149ba3cb2007535b9fd5d9025d2ec9c05a53af 499638

hello\_2.4.orig.tar.gz

90b53e94c626cbd66c48bea2a3668a3cccbc24c943ae9eaa508b153e2abae7d88be3cb33fb166596e7d87e7fb6762367d4d26bc8ba0683a09b7429440c0cf12f 5684 hello\_2.4-3.diff.gz

Package: hello

Binary: hello

Version: 2.7-2

## ● パッケージ情報ファイルの内容(続き)

途中省略

Package: hello

Binary: hello

Version: 2.8-2

途中省略

```
0bd3f47ce5745c7132f63ba5f24289e69947a6c873fd7510b15dfa20ec3adfb8b8c0c6bd
8dbec10a4421448de7eaefaa02653ac1c48c8ca437635d6f1c17785 6540 hello_2.8-
2.debian.tar.gz
```

```
/data1/ubuntu/tests/repo$ cat dists/binary-amd64/Packages
```

Package: hello

Priority: optional

Section: devel

Installed-Size: 648

Maintainer: Ubuntu Developers <ubuntu-devel-discuss@lists.ubuntu.com>

Original-Maintainer: Santiago Vila <sanvila@debian.org>

Architecture: amd64

Version: 2.4-3

Depends: libc6 (>= 2.3.4), dpkg (>= 1.15.4) | install-info

Filename: .././packages/hello\_2.4-3/hello\_2.4-3\_amd64.deb

## ● パッケージ情報ファイルの内容(続き)

Size: 34266

MD5sum: be50fa1d9dff90840a36dd487fcc6195

SHA1: f7f62b6f9840aa7ee3de46fc59290c2667766d4c

SHA256: e7229d3c2e54f55378b315ed4b307cb679064688df081fe9afd0561d6c1d4c96

SHA512:

f4f6f792c32629cbba9eed151976d45bdc6dd23bcff1f40803497ad1b3a818be9034e281

79189ae561d0676f8e69628fdefb9fad0c1f329a9c1befecf9f1e740

Description: The classic greeting, and a good example

The GNU hello program produces a familiar, friendly greeting. It allows non-programmers to use a classic computer science tool which would otherwise be unavailable to them.

.

Seriously, though: this is an example of how to do a Debian package.

It is the Debian version of the GNU Project's `hello world' program (which is itself an example for the GNU Project).

Package: hello

Priority: optional

Section: devel

Installed-Size: 100

Maintainer: Ubuntu Developers <ubuntu-devel-discuss@lists.ubuntu.com>

## ● パッケージ情報ファイルの内容(続き)

Original-Maintainer: Santiago Vila <sanvila@debian.org>

Architecture: amd64

Version: 2.7-2

途中省略

Package: hello

Priority: optional

Section: devel

Installed-Size: 104

Maintainer: Ubuntu Developers <ubuntu-devel-discuss@lists.ubuntu.com>

Original-Maintainer: Santiago Vila <sanvila@debian.org>

Architecture: amd64

Version: 2.8-2

途中省略

```
/data1/ubuntu/tests/repo$ cd
```

```
$
```



- apt-line の編集

リポジトリ構成に合わせて apt-line を編集し、情報を更新する。

```
$ cat /etc/apt/sources.list
deb file:/data1/ubuntu/tests/repo/dists/binary-amd64 ./
deb-src file:/data1/ubuntu/tests/repo/dists/source ./
$ sudo apt-get update
Ign file: ./ InRelease
Ign file: ./ InRelease
Get:1 file: ./ Release.gpg [287 B]
Get:2 file: ./ Release.gpg [287 B]
Get:3 file: ./ Release [1202 B]
Get:4 file: ./ Release [1194 B]
Ign file: ./ Translation-en
Reading package lists... Done
$
```

## ● 更新前のパッケージ情報

```
$ apt-cache show hello
Package: hello
Priority: optional
Section: devel
Installed-Size: 108
Maintainer: Ubuntu Developers <ubuntu-devel-discuss@lists.ubuntu.com>
Original-Maintainer: Santiago Vila <sanvila@debian.org>
Architecture: amd64
Version: 2.8-4
Depends: libc6 (>= 2.14), dpkg (>= 1.15.4) | install-info
Filename: pool/main/h/hello/hello_2.8-4_amd64.deb
Size: 28148
MD5sum: af651f8b526976269affa5b888e63db7
SHA1: c353f6e99dad0dc628b5604741bfbf97eaf74bcf
SHA256: 0cd79de61001c8f6ab7611ba724f0f7da7fd0998bf0fd98710fe26d81f0a6b18
Description-en: The classic greeting, and a good example
The GNU hello program produces a familiar, friendly greeting. It
allows non-programmers to use a classic computer science tool which
would otherwise be unavailable to them.
以下省略
$
```

## ● 更新後のパッケージ情報

```
$ apt-cache show hello
```

```
Package: hello
```

```
Priority: optional
```

```
Section: devel
```

```
Installed-Size: 104
```

```
Maintainer: Ubuntu Developers <ubuntu-devel-discuss@lists.ubuntu.com>
```

```
Original-Maintainer: Santiago Vila <sanvila@debian.org>
```

```
Architecture: amd64
```

```
Version: 2.8-2
```

```
Depends: libc6 (>= 2.14), dpkg (>= 1.15.4) | install-info
```

```
Filename: ./packages/hello_2.8-2/hello_2.8-2_amd64.deb
```

途中省略

```
Package: hello
```

```
Priority: optional
```

```
Section: devel
```

```
Installed-Size: 100
```

```
Maintainer: Ubuntu Developers <ubuntu-devel-discuss@lists.ubuntu.com>
```

```
Original-Maintainer: Santiago Vila <sanvila@debian.org>
```

```
Architecture: amd64
```

```
Version: 2.7-2
```

```
Depends: libc6 (>= 2.4), dpkg (>= 1.15.4) | install-info
```

```
Filename: ./packages/hello_2.7-2/hello_2.7-2_amd64.deb
```

- 更新後のパッケージ情報(続き)

途中省略

Package: hello

Priority: optional

Section: devel

Installed-Size: 648

Maintainer: Ubuntu Developers <ubuntu-devel-discuss@lists.ubuntu.com>

Original-Maintainer: Santiago Vila <sanvila@debian.org>

Architecture: amd64

Version: 2.4-3

Depends: libc6 (>= 2.3.4), dpkg (>= 1.15.4) | install-info

Filename: packages/hello\_2.4-3/hello\_2.4-3\_amd64.deb

途中省略

\$

## ● レポジトリを使ったパッケージ操作

### ➤ パッケージ hold の例

```
$ sudo apt-get install hello=2.4-3
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  hello
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 0 B/34.3 kB of archives.
After this operation, 664 kB of additional disk space will be used.
Selecting previously unselected package hello.
(Reading database ... 97376 files and directories currently installed.)
Unpacking hello (from .../hello_2.4-3_amd64.deb) ...
Processing triggers for install-info ...
Processing triggers for man-db ...
Setting up hello (2.4-3) ...
$ dpkg -l | grep hello
ii hello                2.4-3                amd64        The classic greeting,
and a good example
$
```

## ➤ パッケージ hold の例(続き)

```
$ dpkg --get-selections | grep hello
hello                                install
$ echo hello hold | sudo dpkg --set-selections
$ dpkg --get-selections | grep hello
hello                                hold
$ sudo apt-get upgrade
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages have been kept back:
 hello
0 upgraded, 0 newly installed, 0 to remove and 1 not upgraded.
$ dpkg -l | grep hello
hi hello                                2.4-3                                amd64    The classic greeting,
and a good example
$
```

## ➤ パッケージ hold の例(続き)

```
$ echo hello install | sudo dpkg --set-selections
$ dpkg --get-selections | grep hello
hello                                install
$ sudo apt-get upgrade                Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages will be upgraded:
  hello
1 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
Need to get 0 B/28.1 kB of archives.
After this operation, 557 kB disk space will be freed.
Do you want to continue [Y/n]?
(Reading database ... 97384 files and directories currently installed.)
Preparing to replace hello 2.4-3 (using .../archives/hello_2.8-2_amd64.deb) ...
Unpacking replacement hello ...
Processing triggers for install-info ...
Processing triggers for man-db ...
Setting up hello (2.8-2) ...
$ dpkg -l | grep hello                ii hello                                2.8-2
amd64    The classic greeting, and a good example
$
```

## ➤ バージョン指定のインストール例

```
$ sudo apt-get install hello=2.7-2
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages will be DOWNGRADED:
 hello
0 upgraded, 0 newly installed, 1 downgraded, 0 to remove and 0 not upgraded.
Need to get 0 B/26.1 kB of archives.
After this operation, 4096 B disk space will be freed.
Do you want to continue [Y/n]?
dpkg: warning: downgrading hello from 2.8-2 to 2.7-2
(Reading database ... 97383 files and directories currently installed.)
Preparing to replace hello 2.8-2 (using ../hello_2.7-2_amd64.deb) ...
Unpacking replacement hello ...
Processing triggers for install-info ...
Processing triggers for man-db ...
Setting up hello (2.7-2) ...
kakuma@ubuntu1310-ryu5:~$ dpkg -l | grep hello
ii hello                2.7-2                amd64        The classic greeting,
and a good example
$
```



## ➤ ソースパッケージのダウンロード例

```
$ apt-get source hello=2.4-3
Reading package lists... Done
Building dependency tree
Reading state information... Done
Need to get 0 B/506 kB of source archives.
gpgv: Signature made Sun Aug 30 20:12:18 2009 JST using RSA key ID 9F1B8B32
gpgv: Can't check signature: public key not found
dpkg-source: warning: failed to verify signature on ./hello_2.4-3.dsc
dpkg-source: info: extracting hello in hello-2.4
dpkg-source: info: unpacking hello_2.4.orig.tar.gz
dpkg-source: info: applying hello_2.4-3.diff.gz
dpkg-source: info: upstream files that have been modified:
 hello-2.4/doc/Makefile.in
$ ls -ACFI
total 16
drwxrwxr-x 11 kakuma kakuma 4096 May 16 09:15 hello-2.4/
lrwxrwxrwx 1 kakuma kakuma 84 May 16 09:15 hello_2.4-3.diff.gz ->
/data1/ubuntu/tests/repo/dists/source/../../packages/hello_2.4-3/hello_2.4-3.diff.gz
lrwxrwxrwx 1 kakuma kakuma 80 May 16 09:15 hello_2.4-3.dsc ->
/data1/ubuntu/tests/repo/dists/source/../../packages/hello_2.4-3/hello_2.4-3.dsc
lrwxrwxrwx 1 kakuma kakuma 86 May 16 09:15 hello_2.4.orig.tar.gz ->
/data1/ubuntu/tests/repo/dists/source/../../packages/hello_2.4-3/hello_2.4.orig.tar.gz
$
```

# パッケージの自動更新

cron-apt パッケージでパッケージの自動更新が可能。

- /etc/cron-apt/config  
環境設定ファイル。自動更新を行う action 指定ディレクトリ(default ACTIONDIR="/etc/cron-apt/action.d")、自動更新を行うコマンド(default APTCOMMAND=/usr/bin/apt-get。aptitude も使用可)等を指定できる。  
詳細は /usr/share/doc/cron-apt/README.gz 参照。  
デフォルトでは何も指定されていない。
- /etc/cron-apt/action.d/3-download  
自動更新 Action の指定。  
デフォルトではダウンロードのみ行う。
- /etc/cron.d/cron-apt  
自動更新日時の指定。デフォルトでは 4時に更新する様に指定。

## ➤ 設定ファイル

```
$ sudo apt-get install cron-apt
Reading package lists... Done
...
$ cat /etc/cron-apt/config
# Configuration for cron-apt. For further information about the possible
# configuration settings see /usr/share/doc/cron-apt/README.gz.

$ cat /etc/cron-apt/action.d/3-download
autoclean -y
dist-upgrade -d -y -o APT::Get::Show-Upgraded=true
$ cat /etc/cron.d/cron-apt
#
# Regular cron jobs for the cron-apt package
#
# Every night at 4 o'clock.
0 4 * * * root test -x /usr/sbin/cron-apt && /usr/sbin/cron-apt
# Every hour.
# 0 * * * * root test -x /usr/sbin/cron-apt && /usr/sbin/cron-apt /etc/cron-apt/config2
# Every five minutes.
# */5 * * * * root test -x /usr/sbin/cron-apt && /usr/sbin/cron-apt /etc/cron-apt/config2
$
```