Arch Linux Packaging

David Runge

2019-10-21

Contents

Outline

Package creation

Contact

Who?

- Trusted User (2017)/ Developer (2019)
- Pro-audio, Python tools, web apps
- Documentation

What?

- How is a package created?
- How are split packages created?
- What is DESTDIR?
- How are dependencies handled?
- How does versioning work?
- Where do packages go once they are created?
- How do frontends get data from the packages?

PKGBUILD

- A PKGBUILD[1] is just bash
- makepkg[2] builds the package script and creates a package
- devtools[3] allow for building in a clean chroot
- Packages are installed with the package manager (pacman[4])

[1] https://jlk.fjfi.cvut.cz/arch/manpages/man/PKGBUILD.5
[2] https://jlk.fjfi.cvut.cz/arch/manpages/man/makepkg.8
[3] https://git.archlinux.org/devtools.git/
[4] https://jlk.fjfi.cvut.cz/arch/manpages/man/core/pacman/pacman.8.en

PKGBUILD (definitions)

```
pkgname=NAME
pkgver=VERSION
pkgrel=1
epoch=
pkgdesc=""
arch=()
url=""
license=('GPL')
groups=()
depends=()
makedepends=()
checkdepends=()
optdepends=()
provides=()
conflicts=()
replaces=()
backup=()
options=()
install=
changelog=
source=("$pkgname-$pkgver.tar.gz"
        "$pkgname-$pkgver.patch")
noextract=()
md5sums=()
validpgpkeys=()
```

PKGBUILD (functions)

```
prepare() {
  cd "$pkgname-$pkgver"
  patch -p1 -i "$srcdir/$pkgname-$pkgver.patch"
}

build() {
  cd "$pkgname-$pkgver"
  ./configure --prefix=/usr
  make
}

check() {
  cd "$pkgname-$pkgver"
  make -k check
}

package() {
  cd "$pkgname-$pkgver"
  make DESTDIR="$pkgdir/" install
}
```

Split packages

- **pkgbase** is used to declare the *base*
- **pkgname** as an array can define more than one *package*
- Defining several *packages* requires several *package_pkgname* functions
- Usually used to split out huge documentation blobs or build for different versions of a given language (e.g. python2/python3)
- Not limited to using the same sources (but that's usually the case)

DESTDIR

- Hystorically DESTDIR[5] is used to define an alternative installation destination during *make install*
- When building a package (which is basically an overlay to the filesystem), this is very important (to not install to the build machine's file system)

[5] https://www.gnu.org/prep/standards/html_node/DESTDIR.html

Dependencies

- The **depends** array tracks direct (runtime) dependencies (naming shared libraries directly is also possible)
- The **makedepends** array tracks dependencies only required for building the software (e.g. git, meson, cmake)
- The **checkdepends** array tracks dependencies only required for testing the software (e.g. cxxtest, python-pytest) after successful build
- The **optdepends** array tracks dependencies only indirectly required at runtime (e.g. to extend the functionality)
- The **provides** array tracks packages, components, or libraries a given package provides (e.g. *somesharedlibrary.so*, *somesubcomponent*)
- All tracking allows for $>=$, $<=$ or $=$ assignment for potential version pinning

Package contents

- **.MTREE** tracks all files being installed to the system
- **.BUILDINFO** tracks all meta information about the package and the build circumstances
- **.PKGINFO** tracks all metadata about the package
- A **.install** performs post installation actions, based on predefined (known) functions (similar to PKGBUILD)
- The files (as an overlay to the root filesystem)

Versioning

- The **pkgver** string tracks the **source** version
- The **pkgrel** string tracks the **package release**
- The **epoch** string is used to downgrade a package (to have a way of overruling the **pkgver**-**pkgrel** combination)

Upload

- Packages and their GPG signatures are uploaded to the package server after build and test
- The dbscripts[6] are adding the package metadata to the package (repository) database
- The package database is updated (downloaded) and used by pacman to update packages

[6] https://git.archlinux.org/dbscripts.git/

Website

- The website periodically imports the latest package database and ingests it
- The website's database allows for querying various features of packages (e.g. package file contents, names, dependencies, packager information)

Contact

David Runge

Mail: dave@sleepmap.de

XMPP: dvzrv@sleepmap.de

IRC: dvzrv@{freenode,hackint,oftc}