How To Use Yaourt to Easily Download Arch Linux Community Packages

Authored by: ***ASPHostServer Administrator*** *[asphostserver@gmail.com]*
Saved From: http://faq.asphosthelpdesk.com/article.php?id=212

Introduction

Something that can discourage Arch Linux newcomers is the lack of official packages downloadable with its standard package manager pacman. In this article, I hope to explain why Arch has chosen to work this way, as well as how you can easily acquire just about any package in it.

Glossary:

- PKGBUILD: Build script for making an Arch package
- makepkg: Tool that reads a PKGBUILD and produces an Arch package
- AUR: Arch User Repository, collection of community PKGBUILDs
- pacman: Standard package management tool for Arch Linux

Philosophy

After creating a fresh Arch Linux server instance, you have access to three standard repositories with pacman:

- core
- extra
- community

The core and extra repositories are maintained by the powers that be. Community is maintained by Arch users, but it might not be what you expect. The Arch maintainers are very strict about what becomes a standard package. In addition to achieving a vote threshold, an official community package must be adopted by what Arch refers to as a "Trusted User." The maintainers do not see a reason for a package to be accessible with the 1st party package manager unless it is likely to be maintained at a 1st party level. This is as opposed to trying to maintain as many packages as possible, which many other distributions have tried (and often failed) to do. As of this writing, there are only 36 Trusted Users and only 2000+ community packages.

However, unlike most other distributions, Arch Linux also keeps an additional community repository where users can play fast and loose. This is the Arch User Repository. The AUR is transparent and onlyn moderated by the collective, so it encourages organic growth. As of this writing, there are 44,000+ packages in the AUR. This is in contrast to 10,000+ standard packages.

AUR packages are not source packages or binary packages. Instead, they are instructions (PKGBUILDs) for acquiring, building (if necessary), and installing software. This way of approaching package management was borrowed from FreeBSD Ports. On the source-binary spectrum, a PKGBUILD might use as a source anything from a git repository to a .deb package. In addition, there are no licensing restrictions, unlike in some repositories.

Arch Linux is a rolling release distribution, so any package you download from the AUR or the standard repositories will be the most up to date one made for Arch, even if you are using an older release of the distro. Since there is only one downloadable version of any given package, packages are easier to fix, and fixes apply for everyone. If an update does break something, downgrading can be as easy as changing the source file version in the PKGBUILD, provided the older source file is still available.

Summary: Arch Linux Vs Mainstream Linux

Arch Linux:

- Standard packages maintained by Arch maintainers and Trusted Users
- Shady community packages kept locked away in AUR realm
- One release of a package at any given time = crowd repair effect
- Transparency encourages community involvement and organic growth

Mainstream Linux:
- Standard packages downloaded equally although not maintained equally
- Package version depends on release = crowd repair effect unfeasible
- Opaque community packaging discourages organic growth

Ubuntu has a repository similar to the AUR called universe. As of 9.04 (Jaunty), the universe repository is enabled by default. To put this in perspective, the situation in Ubuntu is analogous to giving Arch users access to several different AURs with pacman right out of the box, not allowing closed-source projects, and not providing a means for immediate community feedback.

Here is the nodejs git package in the AUR

Here is a nodejs dev package in Ubuntu's Universe repository

In addition to the lack of a conversation about the package, notice what words appear on the side of the Ubuntu package page:

"It should generally not be necessary for users to contact the original maintainer."

French for Yogurt

Unfortunately, even though there are many fantastic and well-maintained packages in the AUR, there is no way to easily access them after a fresh Arch install. However, the ArchWiki has a page on many AUR helpers which simplify the acquisition of packages. One of the best of these helpers is a tool by archlinux.fr called yaourt.

Yaourt reduces the cumbersome process of installing AUR packages down to a keyword search and a series of yes or no questions. It is also effectively a replacement for pacman since it can search for and install both AUR and standard packages.

Download Egg, Hatch Chicken

Yaourt can be acquired in a couple of ways:

1. Add the archlinuxfr repository to /etc/pacman.conf
2. Build the yaourt package from the AUR

I will cover both options, but the first is far simpler. If you would like to know how downloading and building packages from the AUR usually works, skip to
method 2.

Method 1: Install via Custom Repository

Open pacman's configuration file to add the custom repository.

```
sudo vi /etc/pacman.conf
```

Add the archlinuxfr repository to the bottom.

```
[archlinuxfr]
SigLevel = Never
Server = http://repo.archlinux.fr/$arch
```

The $arch variable just holds whether your distribution is x86 or x86_64. It will be replaced when the file is processed.

After the custom repository has been added to pacman.conf, the package database needs to be synchronized to include packages from archlinuxfr and any updates from standard packages that yaourt or its dependencies need.

```
sudo pacman -Sy
```

Now, yaourt and its dependencies can be installed just like any other package via pacman.

```
sudo pacman -S yaourt
```

Alternatively, you can do both in one line.

```
sudo pacman -Sy yaourt
```

Method 2: Install via AUR

Although not for the faint of heart, this method is actually how you would normally download, build, and install any package from the AUR in the absence of a helper tool like yaourt.

The general process goes like this:

1. download tarball containing PKGBUILD
2. expand tarball
3. run makepkg in the folder containing PKGBUILD
4. install the .xz file that is produced

First, yaourt needs the package-query package in order to work.

Make sure the standard repositories are up to date for anything that package-query or yaourt might need.

```
sudo pacman -Sy
```

Pull the file from the AUR.

```
curl -O https://aur.archlinux.org/packages/pa/package-query/package-query.tar.gz
```

Expand the file to get access to the PKGBULD.

```
tar zxvf package-query.tar.gz
```

Enter the folder containing the PKGBUILD.

```
cd package-query
```

Make the package. The -s flag syncs the package's standard repository dependencies (if it has any) with pacman before trying to build the package.

```
makepkg -s
```

**Alternative: If you do not have a user set up, add --asroot to force makepkg to run as root. Under normal circumstances you should not build packages
with root permissions, since a PKGBUILD could contain malicious or erroneous code.**

```
makepkg -s --asroot
```

Install the .xz file produced by makepkg. As a shortcut, you can run makepkg with -i (i.e. makepkg -si instead of makepkg -s) to include this step.

```
sudo pacman -U *.xz
```

Next, do the same with the yaourt tarball.

```
curl -O https://aur.archlinux.org/packages/ya/yaourt/yaourt.tar.gz
tar zxvf yaourt.tar.gz
cd yaourt
makepkg -si
```

Note: Even though package-query is a dependency of yaourt, giving makepkg the -s flag won't sync it since it is not in the standard repositories. That is why it needed to be built separately. Although package-query is in the archlinuxfr repository, once you add that, you might as well install yaourt through pacman. Since yaourt handles AUR dependencies, you will not normally need to build AUR dependencies separately.

Using Yaourt

Typical yaourt use starts with passing it a desired keyword. Yaourt will look in both package names and descriptions.

```
yaourt <keyword>
```

When you perform a search, yaourt lists for each matching package:

- description
- version number
- whether the package is installed
- votes on the package

If an installed package is older than the one in the AUR, it will be highlighted. This can be helpful for tracking down packages that are breaking your system.

When you pick an package from the list, yaourt shows the latest comments and asks if you would like to edit the PKGBUILD. This allows you to edit the package in place if something is wrong with it. If someone has posted a fix in the comments, you can make the changes to the PKGBUILD before yaourt runs it through makepkg.

In addition to search, yaourt supports other standard package management operations.

- yaourt -S : install or update a package
- yaourt -Sy: synchronize the pacman package database
- yaourt -R : remove a package
- yaourt -G : get the PKGBUILD for a package
- yaourt --stats: show how much space packages are using

Yaourt has an export option for storing packages after they have been built.

```
yaourt -Sb --export <destination dir> <package>
```

This could be used for making binary backups of critical packages. If you want to build and archive the package without installing it, just say no at the installation step. Since yaourt is an interactive tool, you will likely want to find something else if you need to automate this process.