AUR submission guidelines

Related articles

Arch User Repository

Arch package guidelines

AUR Trusted User Guidelines

Users can **share** PKGBUILDs using the **Arch User Repository**. It does not contain any binary packages but allows users to upload PKGBUILDs that can be downloaded by others. These PKGBUILDs are completely unofficial and have not been thoroughly vetted, so they should be used at your own risk.

Contents

Submitting packages

Warning: Before attempting to submit a package you are expected to familiarize yourself with **Arch packaging standards** and all the articles under "Related articles". **Verify carefully** that what you are uploading is correct. Packages that violate the rules may be **deleted** without warning.

If you are unsure in any way about the package or the build/submission process even after reading this section twice, submit the PKGBUILD to the **AUR mailing list (https://mailman.archlinux.org/mailman/listinfo/aur-general)**, the **AUR forum (https://bbs.archlinux.org/viewforum.php?id=4)** on the Arch forums, or ask on our **IRC channel** for public review before adding it to the AUR.

Rules of submission

When submitting a package to the AUR, observe the following rules:

- The submitted PKGBUILDs must not build applications **already in any** of the **official** binary **repositories** under any circumstances. Check the **official package database (https://archlinux.org/packages/)** for the package. If any version of it exists, **do not** submit the package. If the official package is out-of-date, flag it as such. If the official package is broken or is lacking a feature, then please file a **bug report (https://bugs.archlinux.org/)**.

  Exception to this strict rule may only be packages having **extra features** enabled and/or **patches** in comparison to the official ones. In such an occasion the `pkgname` should be different to express that difference. For example, a package for GNU screen containing the sidebar patch could be named `screen-sidebar` . Additionally the `provides=('screen')` array should be used in order to avoid conflicts with the official package.

- **Check the AUR** if the package **already exists**. If it is currently maintained, changes can be submitted in a comment for the maintainer's attention. If it is unmaintained or the maintainer is unresponsive, the package can be adopted and updated as required. Do not create duplicate packages.

- Make sure the package you want to upload is **useful**. Will anyone else want to use this package? Is it extremely specialized? If more than a few people would find this package useful, it is appropriate for submission.

  The AUR and official repositories are intended for packages which install generally software and software-related content, including one or more of the following: executable(s); config file(s); online or offline documentation for specific software or the Arch Linux distribution as a whole; media intended to be used directly by software.

- Do not use `replaces` in an AUR PKGBUILD unless the package is to be renamed, for example when *Ethereal* became *Wireshark*. If the package is an **alternate version of an already existing package**, use `conflicts` (and `provides` if that package is required by others). The main difference is: after syncing (-Sy) pacman immediately wants to replace an installed, 'offending' package upon encountering a package with the matching `replaces` anywhere in its repositories; `conflicts`, on the other hand, is only evaluated when actually installing the package, which is usually the desired behavior because it is less invasive.

- Packages that use **prebuilt deliverables**, when the sources are available, must use the `-bin` suffix. An exception to this is with Java. The AUR should not contain the binary tarball created by makepkg, nor should it contain the filelist.

- Please add a **comment line** to the top of the `PKGBUILD` file which contains information about the current **maintainers** and previous **contributors**, respecting the following format. Remember to disguise your email to protect against spam. Additional or unneeded lines are facultative.

  If you are assuming the role of maintainer for an existing PKGBUILD, add your name to the top like this

  ```
  # Maintainer: Your Name <address at domain dot tld>
  ```

  If there were previous maintainers, put them as contributors. The same applies for the original submitter if this is not you. If you are a co-maintainer, add the names of the other current maintainers as well.

  ```
  # Maintainer: Your name <address at domain dot tld>
  # Maintainer: Other maintainer's name <address at domain dot tld>
  # Contributor: Previous maintainer's name <address at domain dot tld>
  # Contributor: Original submitter's name <address at domain dot tld>
  ```

Authentication

For write access to the AUR, you need to have an SSH key pair. The content of the public key needs to be copied to your profile in *My Account*, and the corresponding private key configured for the `aur.archlinux.org` host. For example:

```
~/.ssh/config
```

```
Host aur.archlinux.org
  IdentityFile ~/.ssh/aur
  User aur
```

You should create a new key pair rather than use an existing one, so that you can selectively revoke the keys should something happen:

```
$ ssh-keygen -f ~/.ssh/aur
```

Tip: You can add multiple public keys to your profile by separating them with a newline in the input field.

Creating package repositories

If you are creating a new package from scratch, establish a local Git repository and an AUR remote by cloning the intended pkgbase. If the package does not yet exist, the following warning is expected:

```
$ git clone ssh://aur@aur.archlinux.org/pkgbase.git
```

```
Cloning into 'pkgbase'...
warning: You appear to have cloned an empty repository.
Checking connectivity... done.
```

Note: The repository will not be empty if `pkgbase` matches a deleted package.

If you already have a package, initialize it as a Git repository if it is not one, and add an AUR remote:

```
$ git remote add label ssh://aur@aur.archlinux.org/pkgbase.git
```

Then fetch this remote to initialize it in the AUR.

Note: Pull and rebase (https://git-scm.com/docs/git-pull#git-pull---rebasefalsetruemergespreserveinteractive) to resolve conflicts if `pkgbase` matches a deleted package.

Publishing new package content

Warning: Your commits will be authored with your global Git name and email address. It is very difficult to change commits after pushing them (FS#45425 (https://bugs.archlinux.org/task/45425)). If you want to push to the AUR under different credentials, you can change them per package with `git config user.name "..."` and `git config user.email "..."` .

When releasing a new version of the packaged software, update the pkgver or pkgrel variables to notify all users that an upgrade is needed. Do not update those values if only minor changes to the PKGBUILD such as the correction of a typo are being published.

Be sure to regenerate .SRCINFO whenever `PKGBUILD` metadata changes, such as `pkgver()` updates; otherwise the AUR will not show updated version numbers.

To upload or update a package, add *at least* `PKGBUILD` and `.SRCINFO` , then any additional new or modified helper files (such as *.install* files or local source files such as patches), commit with a meaningful commit message, and finally push the changes to the AUR.

For example:

```
$ makepkg --printsrcinfo > .SRCINFO
$ git add PKGBUILD .SRCINFO
$ git commit -m "useful commit message"
$ git push
```

Note: If `.SRCINFO` was not included in your first commit, add it by **rebasing with --root (https://git-scm.com/docs/git-rebase#git-rebase---root)** or **filtering the tree (https://git-scm.com/docs/git-filter-branch#git-filter-branch---tree-filterltcommandgt)** so the AUR will permit your initial push.

Tip: To keep the working directory and commits as clean as possible, create a **`gitignore(5) (https://man.archlinux.org/man/gitignore.5)`** that excludes all files and force-add files as needed.

Maintaining packages

- Check for feedback and comments from other users and try to incorporate any improvements they suggest; consider it a learning process!
- Please do not leave a comment containing the version number every time you update the package. This keeps the comment section usable for valuable content mentioned above.
- Please do not just submit and forget about packages! It is the maintainer's job to maintain the package by checking for updates and improving the PKGBUILD.
- If you do not want to continue to maintain the package for some reason, `disown` the package using the AUR web interface and/or post a message to the AUR Mailing List. If all maintainers of an AUR package disown it, it will become an **"orphaned" (https://aur.archlinux.org/packages/?SB=n&do_Orphans=Orphans)** package.

Requests

Orphan, deletion and merge requests can be created by clicking on the "Submit Request" link under "Package Actions" on the right hand side. This automatically sends a notification email to the current package maintainer and to the **aur-requests mailing list (https://mailman.archlinux.org/mailman/listinfo/aur-requests)** for discussion. **Trusted Users** will then either accept or reject the request.

- Orphan requests will be granted after two weeks if the current maintainer did not react. The exception is if a package was flagged out-of-date for at least 180 days; orphan requests are then automatically accepted.
- Merge requests are to delete the package base and transfer its votes and comments to another package base. The name of the package base to merge into is required. Note this has nothing to do with 'git merge' or GitLab's merge requests.
- Deletion requests require the following information:
  - A short note explaining the reason for deletion. Note that a package's comments does not sufficiently point out the reasons why a package is up for deletion. Because as soon as a TU takes action, the only place where such information can be obtained is the aur-requests mailing list.
  - Supporting details, like when a package is provided by another package, if you are the maintainer yourself, it is renamed and the original owner agreed, etc.
  - After a package is deleted, its Git repository remains available in the AUR.

Retrieved from "**https://wiki.archlinux.org/index.php?title=AUR_submission_guidelines&oldid=717038**"

This page was last edited on 7 February 2022, at 10:54.

- Privacy policy
- About ArchWiki
- Disclaimers